Scientific Excellence • Resource Protection & Conservation • Benefits for Canadians
Excellence scientifique • Protection et conservation des ressources • Bénéfices aux Canadiens

Compiler Listing for Chemical Speciation Program MACS80.VERS.4.1 for Microsoft® Fortran Version 5 for MS-DOS®

R. Wagemann, G. Regehr and R. Lypka

Central and Arctic Region
Department of Fisheries and Oceans
Winnipeg, Manitoba R3T 2N6

December 1990

Canadian Data Report of Fisheries and Aquatic Sciences No. 818

Fisheries and Oceans
Pêches et Océans

Canada

Canadian Data Report of Fisheries and Aquatic Sciences

Data reports provide a medium for filing and archiving data compilations where little or no analysis is included. Such compilations commonly will have been prepared in support of other journal publications or reports. The subject matter of data reports reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Data reports are not intended for general distribution and the contents must not be referred to in other publications without prior written authorization from the issuing establishment. The correct citation appears above the abstract of each report. Data reports are abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1-25 in this series were issued as Fisheries and Marine Service Data Records. Numbers 26-160 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Data Reports. The current series name was introduced with the publication of report number 161.

Data reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport statistique canadien des sciences halieutiques et aquatiques

Les rapports statistiques servent à classer et à archiver les compilations de données pour lesquelles il y a peu ou point d'analyse. Ces compilations auront d'ordinaire été préparées à l'appui d'autres publications ou rapports. Les sujets des rapports statistiques reflètent la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports statistiques ne sont pas destinés à une vaste distribution et leur contenu ne doit pas être mentionné dans une publication sans autorisation écrite préalable de l'établissement auteur. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports statistiques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annuel des publications scientifiques et techniques du Ministère.

Les numéros 1 à 25 de cette série ont été publiés à titre de relevés statistiques, Services des pêches et de la mer. Les numéros 26 à 160 ont été publiés à titre de rapports statistiques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 161.

Les rapports statistiques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

91-02558/

Canadian Data Report of
Fisheries and Aquatic Sciences 818

December 1990

COMPILER LISTING FOR CHEMICAL SPECIATION PROGRAM MACS80.VERS.4.1 FOR MICROSOFT® FORTRAN VERSION 5 FOR MS-DOS®

by

R. Wagemann, G. Regehr and R. Lypka

Central and Arctic Region
Department of Fisheries and Oceans
Winnipeg Manitoba R3T 2N6

This is the 53rd Data Report
from the Central and Arctic Region

(c) Minister of Supply and Services Canada 1990

Cat. no. Fs 97-13/818E ISSN 0706-6465

Correct citation for this publication is:

Wagemann, R., G. Regehr, and R. Lypka. 1990. Compiler listing for chemical speciation program MACS80.VERS.4.1 for Microsoft® Fortran Version 5 for MS-DOS®. Can. Data Rep. Fish. Aquat. Sci. 818: x+190 p.

TABLE OF CONTENTS

Page

ABSTRACT/RÉSUMÉ iv

INTRODUCTION v

ACKNOWLEDGMENTS vii

MACS80.INP FILE viii

PROGRAM LISTING x

ABSTRACT

Wagemann, R., G. Regehr, and R. Lypka. 1990. Compiler listing for chemical speciation program MACS80.VERS.4.1 for Microsoft Fortran Version 5 for MS-DOS. Can. Data Rep. Fish. Aquat. Sci. 818: x+190 p.

A listing of the MACS80.VERS.4.1 chemical speciation program for fresh water is given. "Seed" values are not required to initiate calculations; only the concentrations of total dissolved anions and metals, i.e. analytical concentrations, ionic strength or conductivity, temperature, pH range and pH increments are required inputs. The program can calculate as a function of pH the concentration of 356 chemical species generated from the interaction of 15 anions and 14 cations. The interaction between dissolved organic matter and copper and cadmium, and between cadmium and suspended sediment is taken into account. Calculations can be performed in equilibrium with one or more of 62 different solids and under "open" or "closed" conditions with respect to exchange of atmospheric CO_2.

Key words: computer programs; chemical speciation; equilibrium constant; fresh water; metals.

RÉSUMÉ

Wagemann, R., G. Regehr, and R. Lypka. 1990. Compiler listing for chemical speciation program MACS80.VERS.4.1 for Microsoft Fortran Version 5 for MS-DOS. Can. Data Rep. Fish. Aquat. Sci. 818: x+190 p.

On donne un listage de la version 4.1 du programme de caractérisation des espèces chimiques MACS80. Des valeurs "de départ" ne sont pas nécessaires pour entreprendre les calculs; les seules entrées nécessaires sont les concentrations des espèces anioniques et métalliques totales en solution, c'est-à-dire les concentrations mesurées par analyse, la force ionique ou la conductivité, la température, la plage de pH et les incréments de pH. Le programme peut calculer, en fonction de pH, la concentration de 356 espèces chimiques produites par l'interaction de 15 anions et 14 cations. Il tient compte également de l'interaction entre le cadmium, et entre le cadmium et les sédiments en suspension. Il peut effecteur les calculs en conditions d'équilibre avec 62 solides et en conditions "ouvertes" ou "fermées" pour ce qui est des échanges de CO_2 avec l'atmosphère.

Mots-clés: programmes informatiques; caractérisation des espéces chimiques; équilibre chimique; eau douce; métaux.

INTRODUCTION

The MACS80 program calculates the concentration of 356 different chemical species resulting from the interaction of 14 metals with 15 anions and their various hydrogen ion-associated forms. A proton mass balance condition is not imposed. pH is a required input. Mass balances for all other ions are imposed. The algorithm is based on the equilibrium constant approach. The resulting sets of nonlinear mass balance equations are expressed as polynomials in terms of cations and anions respectively and are solved reiteratively by going back and forth between the cation and anion polynomials. The program has the unique feature of not requiring estimated starting values. It generates its own appropriate starting values by performing initially a calculation for the hydroxy-metal complexes only, having obtained the activity of the hydroxide ion from the given pH. This yields an initial cation concentration which is used in the anion polynomials to obtain an improved anion concentration which in turn is used in the cation polynomials to obtain a better cation estimate and so on. The reiterative process is carried on until the defined precision limit is reached for each ion (0.01%) between two successive reiterations, or until 100 reiterations are exceeded. The iterations almost always converge. Very rarely does the program "bomb". If this happens it is usually not for lack of convergence but because of extreme values of the coefficients in the polynomial, which then makes it difficult to obtain meaningful roots. The two routines that are used to obtain the roots of polynomials are the explicit solution of the quadratic equation in the case of a second degree polynomial, and the bisection method for higher degree polynomials. The highest degree polynomial presently in the program is 3. For second degree polynomials the explicit solution is always used first since it is much faster, but if the roots cannot be found in this way because of extreme values of the coefficients in the quadratic equation the second method is used.

The executable file is named "MACS80". The required input consists of the specification of the concentrations of the 28 ions (the concentration of the hydroxide ion is not required) in either of two units, mg/L or M, and values for ionic strength or conductivity, temperature etc. in an input file named "MACS80.INP". This input file can be edited with any ascii text editor. Storing the program on disk requires 734 Kb, but it will run on computers with only 640 Kb RAM memory since the program consists of a number of overlays. Outputs can be obtained for each anion and cation in the form of a table listing all the chemical species concentrations involving that ion, by indicating true or false opposite that ion under "REPORT" in the input file. The species concentration is calculated at each incremental pH value within the pH range specified by the pH minimum and maximum for all the requested species and put in a file called "MACS80.OUT". The output concentrations can be obtained in one or more of six different units: (1)-M, (2)-logM, (3)-%M, (4)-mg/L, (5)-log(mg/L), (6)-%(mg/L) by indicating true or false in the appropriate place under "CONCENTRATION" in the input file. Each output table can be copied to an ascii data file by providing a file name in the column "FILE NAME" in the input file opposite the ion in question. The extension ".DAx" is appended in this file name where "x" is a number between 1 and 6, and defines the units of the output i.e. 1=M, 2=logM,...etc. Graphs of pH vs concentration of any species, in any of the specified units, can then be plotted from the values stored in the ".DAx" files.

The program can also calculate the concentration of species in equilibrium with any of 62 different solids. To perform calculations under these conditions a number, identifying the particular solid, is entered opposite the anion or cation of that solid in the "SOLID" column of the input file. Only one of the ions in question is allowed to vary in concentration as a function of pH to satisfy the requirement of the solubility equation. When the solid number is entered opposite the anion, the anion concentration is allowed to vary, and the cation concentration remains constant at the inputted value; similarly, if the solid number is entered opposite the cation, the cation concentration is allowed to vary, and the anion concentration remains constant.

The program can calculate the concentration of chemical species under "open" and "closed" conditions with respect to atmospheric CO_2 transport in and out of the system. Open system conditions are invoked, i.e. the total dissolved inorganic carbon concentration in the system is allowed to vary as a function of pH by setting the "CONTROLS T.D. IC" parameter to "T" in the input file. Closed system conditions are invoked i.e. the total dissolved inorganic carbon concentration remains fixed at the input value at all pH levels by setting this parameter to "F".

The program takes into account the interaction between dissolved organic matter and copper and cadmium and calculates the concentration of the corresponding organo-metal complexes. For cadmium the interaction with suspended or bottom sediment has been incorporated. The sediment adsorption constant for cadmium was determined experimentally using ^{109}Cd and sediments from a "shield" lake in the "Experimental Lakes Area", Ontario, Canada. this constant is a function of pH and sediment composition. It can be easily changed from the given value by inputting a different value in the **MACS80.INP FILE** under "SEDIMENT CD CONSTANT=". To change other constants their values must be changed within the program.

A list of all equilibrium constants, activity coefficients and solids and their code numbers can be obtained by setting the "DEFINITIONS" parameter to "T" in the input file. Equilibrium constants marked with an asterisk in this table are adjusted for temperature if the specified temperature is different from 25°C. Constants not marked in this fashion are only known for 25°C since the necessary enthalpy of complexion data (ΔH) that would allow expressing them as a function of temperature were not readily available. Activity coefficients are calculated with the Debye-Hückel equation, taking into account the size of the ions and are valid for ionic strengths of 0.1 M and below.

ACKNOWLEDGMENTS

Development of the MACS80 program began in 1977 for the AMDAHL mainframe, and proceeded at intervals through a number of different versions involving refinements and the incorporation of additional metals and anions. At different times a number of programmers assisted in these early efforts to whom we owe our sincerest thanks.

D. Abrams
W. Elliot
M. Judd
D. Bayomi

The program has recently been translated by one of us (G. Regehr) into Microsoft Fortran, Version 5.0 to run on MS-DOS microcomputers.

MACS80.INP FILE

```
TITLE                   STANDARD RIVER WATER TEST DATA

TEMPERATURE             25.0

PCO2                    0.000350   CONTROLS T.D. IC=F

IONIC STRENGTH          0.0036

CONDUCTIVITY            0.0

pH RANGE            MIN=4.0    MAX=9.5    INC=0.5
pH SPECIFIC             8.01

SEDIMENT CD CONSTANT =  5.43515D-5 *10^( .5064    *PH)
OUTPUT UNITS:        MOLES/L  MG/L

     CONCENTRATION      F      T
         LOG(CONC)      T      F
           %(CONC)      F      F
DEFINITIONS             F
```

MACS80.INP FILE (cont'd)

```
OUTPUT FILE     >MACS<    ID >RUN2<          SOLID    REPORT  FILENAME
A SULFATE              2  2.57       MG/L       00         F    >      <
A EDTA                 3  0          MG/L       00         F    >      <
A NTA                  4  0          MG/L       00         F    >      <
A INORGANIC CARBON     5  14.8       MG/L       00         F    >      <
A CHLORIDE             6  9.9        MG/L       00         F    >      <
A ORGANIC CARBON       7  2.5        MG/L       00         T    >      <
A SELENITE             8  2D-4       MG/L       00         F    >      <
A SILICATE             9  8.52       MG/L       00         F    >      <
A AMMONIA             10  .112       MG/L       00         F    >      <
A ORTHO-PHOSPHATE     11  0.21       MG/L       00         F    >      <
A ACETATE             12  0          MG/L       00         F    >      <
A SULFIDE             13  2.0D-3     MG/L       00         T    >      <
A SEDIMENT            14  1.0        MG/L       00         T    >      <
A FLUORIDE            15  0.0        MG/L       00         F    >      <
M SODIUM               1  12.0       MG/L       00         F    >      <
M MAGNESIUM            2  7.5        MG/L       00         F    >      <
M CALCIUM              3  12.2       MG/L       00         F    >      <
M IRON(III)            4  7D-3       MG/L       00         F    >      <
M MERCURY(I)           5  0          MG/L       00         F    >      <
M MERCURY(II)          6  1D-5       MG/L       00         F    >      <
M CADMIUM              7  1D-4       MG/L       00         T    >      <
M COPPER               8  5D-4       MG/L       00         T    >      <
M IRON(II)             9  1.5D-2     MG/L       00         F    >      <
M MANGANESE(II)       10  4.4D-3     MG/L       00         F    >      <
M ZINC                11  4.9D-4     MG/L       00         F    >      <
M POTASSIUM           12  1.4        MG/L       00         F    >      <
M COBALT              13  0          MG/L       00         F    >      <
M ALUMINUM            14  0.0        MG/L       00         F    >      <
```

Fields reserved for inputting data and parameters in the input file must be maintained at their predefined positions.

PROGRAM LISTING

```
C  $DEFINE DEVELOP
      PROGRAM MAIN
C
C                 MACS80-4.FOR
C                 *************
C  CHEMICAL SPECIATION PROGRAM (PC VERSION 4.0) 1990
C     COMPILED WITH MICROSOFT FORTRAN V5.0
C     LARGE MODEL, EMULATOR LIBRARY, INLINE CODE
C
C     D E C L A R A T I O N S
C     -----------------------
C
C  OPTIONS, SPECIAL CONDITIONS AND EQUILIBRIUM FLAGS.
C
      IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
      LOGICAL*2 EQUILB,S$PCO2,S$DEFNS
      LOGICAL*2 S$ASPEC(15),S$MSPEC(14)
      INTEGER UNITS(6)
C
      INTEGER NAVALS/15/,NMVALS/14/,NGVALS/319/,NKVALS/356/
      INTEGER METAL,MAXITR/100/
      REAL*8 AO(319)
      REAL*8 DEXP,DABS,DSQRT
C
C
      REAL*8 TEMP,PCO2,IONIC,CONDUC,PHMIN,PHMAX,PHINC,PHSP
      REAL*8 A(15),MTL(14),SSCDA,SSCDB
      COMMON /IN/TEMP,PCO2,IONIC,CONDUC,
     &   PHMIN,PHMAX,PHINC,PHSP,A,MTL,S$PCO2,SSCDA,SSCDB
C
C  TOTAL DISSOLVED VALUES AND SPECIES CONVERSION FACTORS.
C
      REAL*8 X(319,41)
      REAL*4 PHS(41),TDSO4(41),TDEDTA(41),TDNTA(41),
     &TDIC(41),TDCL(41),TDOC(41),TDSE(41),TDSIO4(41),TDNH3(41),
     &TDPO4(41),TDAC(41),TDS(41),TDNA(41),TDMG(41),TDCA(41),TDFE3(41),
     &TDHG1(41),TDHG2(41),TDCD(41),TDCU(41),TDFE2(41),TDMN2(41),
     &TDZN(41),TDK(41),TDCO(41),TDSS(41),TDF(41),TDAL(41)
      COMMON /OUT/ X,PHS
      COMMON /OUT1/ TDSO4,TDEDTA,TDNTA,TDIC,TDCL,TDOC,
     &              TDSE,TDSIO4,TDNH3,TDPO4,TDAC,TDS,TDSS,TDF,
     &              TDNA,TDMG,TDCA,TDFE3,TDHG1,TDHG2,TDCD,TDCU,
     &              TDFE2,TDMN2,TDZN,TDK,TDCO,TDAL
C  ACTIVITY AND STABILITY CONSTANTS.
C
      INTEGER*2 Z(319)
      REAL*8 C(356),M(356),ALPHA(356),KA(356),GAM(319)
      CHARACTER KTAG(356)*2
      COMMON /OUT2/C,M,ALPHA,KA,GAM,KTAG
C
C  ENVIRONMENTAL CONDITIONS.
```

```
C
      CHARACTER*15 AFILES(15),MTLFILES(14)
      CHARACTER FILEID*4
      INTEGER ASLD(15),MTLSLD(14)
      REAL*8 CONVTA(15),CONVTM(14)
      COMMON /BOTH/CONVTA,CONVTM,ASLD,MTLSLD,UNITS,
     &             FILEID,AFILES,MTLFILES,S$DEFNS,S$ASPEC,S$MSPEC
C
C  INDIVIDUAL SPECIES AND TOTAL CONCENTRATIONS (AQUEOUS AND SOLID PHASE
C
      REAL*8 SPSUM(15,14,9),HSP(15),HSPQ(15),
     &       XANIN(15),XMTL(14),XPREV(14)
      COMMON /CALC/SPSUM,HSP,HSPQ,XANIN,XMTL,XPREV
C
      CHARACTER SLDID*27
      REAL*8 F,F2,F3
      INTEGER PR,NUMPH,NPL
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      REAL*8 TOL/1D-4/,BETA
      CHARACTER*60 BLANKS
      CHARACTER FF
      DATA FF/12/
C
      DATA SPSUM /1890*0D0/
      DATA HSP /15*0D0/
      DATA HSPQ /15*0D0/
      DATA PR /6/

      DATA CONVTA
     &   /1.0000D00, 3.2064D04, 1.2011D05, 7.2067D04, 1.2011D04,
     &    3.5453D04, 1.2011D04, 7.8960D04, 2.8086D04, 1.4007D04,
     &    3.0974D04, 2.4022D04, 3.2064D04, 1.0000D00, 1.8998D04/
      DATA CONVTM
     &   /2.2990D04, 2.4312D04, 4.0080D04, 5.5847D04, 4.0118D05,
     &    2.0059D05, 1.1240D05, 6.3540D04, 5.5847D04, 5.4938D04,
     &    6.5380D04, 3.9098D04, 5.8933D04, 2.6982D04/
C
      DATA (Z(I),I=1,50)
     &   / 2, 1, 2, 0, 2, 2, 0, 0, 0, 0,
     &     1, 1, 1, 1, 1, 1, 1, 2, 0, 2,
     &     1, 0, 1, 2, 2, 0, 1, 1, 0, 0,
     &     2, 2, 2, 1, 0, 1, 0, 1, 0, 2,
     &     1, 0, 1, 2, 3, 0, 1, 0, 1, 2/
      DATA (Z(I),I=51,100)
     &   / 2, 2, 0, 1, 2, 1, 0, 1, 0, 1,
     &     2, 3, 0, 1, 0, 1, 1, 0, 1, 0,
     &     1, 2, 1, 0, 1, 2, 0, 1, 1, 0,
     &     1, 0, 1, 1, 0, 1, 3, 3, 0, 1,
     &     0, 1, 2, 2, 2, 1, 0, 1, 0, 0/
      DATA (Z(I),I=101,150)
```

```
  103       &   / 2, 0, 2, 2, 1, 0, 2, 3, 2, 2,
  104       &     3, 1, 1, 0, 1, 2, 1, 1, 2, 2,
  105       &     2, 1, 0, 1, 3, 0, 1, 0, 2, 4,
  106       &     1, 0, 1, 1, 0, 1, 2, 2, 2, 0,
  107       &     0, 1, 1, 0, 0, 1, 1, 0, 1, 0/
  108         DATA (Z(I),I=151,200)
  109       &   / 2, 2, 1, 4, 1, 1, 2, 1, 1, 2,
  110       &     1, 0, 2, 1, 0, 2, 0, 2, 3, 6,
  111       &    12,10, 2, 1, 3, 2, 1, 4, 2, 2,
  112       &     1, 1, 1, 3, 1, 4, 1, 0, 4, 3,
  113       &     2, 1, 0, 3, 2, 1, 2, 1, 1, 0/
  114         DATA (Z(I),I=201,250)
  115       &   / 2, 3, 4, 2, 1, 3, 2, 1, 2, 1,
  116       &     3, 2, 1, 3, 4, 2, 1, 2, 1, 3,
  117       &     3, 2, 1, 2, 1, 0, 1, 2, 3, 0,
  118       &     2, 2, 2, 0, 1, 3, 2, 1, 0, 2,
  119       &     1, 1, 4, 0, 3, 1, 2, 1, 1, 4/
  120         DATA (Z(I),I=251,300)
  121       &   / 2, 1, 4, 2, 1, 4, 0, 2, 1, 4,
  122       &     1, 4, 2, 1, 4, 2, 0, 1, 0, 0,
  123       &     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  124       &     0, 0, 0, 3, 2, 1, 0, 1, 4, 1,
  125       &     0, 2, 0, 0, 1, 2, 2, 2, 1, 1/
  126         DATA (Z(I),I=301,319)
  127       &   / 1, 2, 1, 0, 0, 1, 0, 2, 1, 0,
  128       &     1, 1, 1, 1, 1, 1, 2, 1, 0/
  129   C
  130         DATA (AO(I),I=1,25)
  131       &   /  4.5D0,  4.0D0,  6.0D0,  1.0D0,  8.0D0,
  132       &      4.5D0,  1.0D0,  1.0D0,  1.0D0,  1.0D0,
  133       &      4.0D0,  4.0D0,  4.0D0,  4.0D0,  4.0D0,
  134       &      3.0D0,  3.5D0,  6.0D0,  1.0D0,  4.5D0,
  135       &      4.0D0,  1.0D0,  4.0D0,  4.5D0,  6.0D0/
  136         DATA (AO(I),I=26,50)
  137       &   /  1.0D0,  4.0D0,  4.0D0,  1.0D0,  1.0D0,
  138       &      4.5D0,  4.5D0,  4.5D0,  2.5D0,  1.0D0,
  139       &      4.0D0,  1.0D0,  4.5D0,  1.0D0,  5.0D0,
  140       &      5.0D0,  1.0D0,  4.5D0,  5.0D0,  5.0D0,
  141       &      1.0D0,  4.0D0,  1.0D0,  5.0D0,  5.0D0/
  142         DATA (AO(I),I=51,75)
  143       &   /  5.0D0,  5.0D0,  1.0D0,  4.5D0,  4.0D0,
  144       &      5.0D0,  1.0D0,  5.0D0,  1.0D0,  3.5D0,
  145       &      5.0D0,  4.0D0,  1.0D0,  4.0D0,  1.0D0,
  146       &      4.5D0,  4.0D0,  1.0D0,  4.5D0,  1.0D0,
  147       &      1.0D0,  6.0D0,  4.5D0,  1.0D0,  4.5D0/
  148         DATA (AO(I),I=76,100)
  149       &   /  4.5D0,  1.0D0,  4.5D0,  4.5D0,  1.0D0,
  150       &      5.0D0,  1.0D0,  5.0D0,  4.5D0,  1.0D0,
  151       &      4.5D0,  4.0D0,  4.0D0,  1.0D0,  4.5D0,
  152       &      1.0D0,  4.5D0,  5.0D0,  5.0D0,  5.0D0,
  153       &      4.5D0,  1.0D0,  4.5D0,  1.0D0,  1.0D0/
```

```
      DATA (A0(I),I=101,125)
     &   / 5.0D0,  1.0D0,  5.0D0,  4.5D0,  1.0D0,
     &     1.0D0,  4.5D0,  4.0D0,  4.5D0,  4.5D0,
     &     5.0D0,  6.0D0,  4.0D0,  1.0D0,  4.0D0,
     &     4.5D0,  4.0D0,  4.0D0,  4.5D0,  4.5D0,
     &     4.5D0,  4.0D0,  1.0D0,  4.0D0,  5.0D0/
      DATA (A0(I),I=126,150)
     &   / 1.0D0,  4.5D0,  1.0D0,  4.5D0,  5.0D0,
     &     4.0D0,  1.0D0,  4.0D0,  4.0D0,  1.0D0,
     &     4.5D0,  4.5D0,  4.5D0,  4.5D0,  1.0D0,
     &     1.0D0,  4.5D0,  4.5D0,  1.0D0,  1.0D0,
     &     4.5D0,  4.5D0,  1.0D0,  4.5D0,  1.0D0/
      DATA (A0(I),I=151,175)
     &   / 5.0D0,  6.0D0,  4.5D0,  6.0D0,  4.5D0,
     &     4.5D0,  5.0D0,  4.5D0,  4.5D0,  6.0D0,
     &     4.5D0,  1.0D0,  5.0D0,  4.5D0,  1.0D0,
     &     5.0D0,  1.0D0,  4.5D0,  6.0D0,  9.0D0,
     &    11.0D0, 11.0D0,  4.5D0,  4.0D0,  6.0D0/
      DATA (A0(I),I=176,200)
     &   / 4.5D0,  4.0D0,  9.0D0,  6.0D0,  6.0D0,
     &     4.0D0,  3.0D0,  4.0D0,  9.0D0,  9.0D0,
     &     5.0D0,  4.5D0,  1.0D0,  6.0D0,  7.5D0,
     &    -0.2D0, -5.1D0,  1.0D0,  0.5D0, -5.8D0,
     &    -8.5D0, -5.9D0, -8.5D0, -8.8D0,  1.0D0/
      DATA (A0(I),I=201,225)
     &   / -6.8D0, -4.1D0,  1.9D0, -5.8D0, -8.5D0,
     &    -2.5D0, -5.9D0, -8.5D0, -5.9D0, -8.5D0,
     &    -0.1D0, -5.9D0, -8.5D0, -2.5D0, -4.1D0,
     &    -5.9D0, -8.5D0, -5.9D0, -8.5D0, -0.1D0,
     &     0.4D0,  6.0D0, -8.5D0, -5.9D0,  4.0D0/
      DATA (A0(I),I=226,250)
     &   / 1.0D0,  4.0D0,  4.5D0,  6.0D0,  1.0D0,
     &     4.5D0,  4.5D0,  4.5D0,  1.0D0,  4.0D0,
     &     0.8D0,  0.8D0,  2.3D0,  1.0D0,10.75D0,
     &   -6.75D0,-6.65D0, 32.9D0,  1.0D0,  5.6D0,
     &    -8.3D0, -7.0D0, -7.1D0, -7.5D0, 32.9D0/
      DATA (A0(I),I=251,275)
     &   / -6.3D0, -7.0D0, 32.9D0,-6.07D0, -7.0D0,
     &    32.9D0,  1.0D0,-6.07D0, -7.0D0, 32.9D0,
     &    -7.0D0, 32.9D0,-6.07D0, -7.0D0, 32.9D0,
     &   -6.07D0,  1.0D0,  4.0D0,  1.0D0,  1.0D0,
     &     1.0D0,  1.0D0,  1.0D0,  1.0D0,  1.0D0/
      DATA (A0(I),I=276,300)
     &    / 1.0D0,  1.0D0,  1.0D0,  1.0D0,  1.0D0,
     &     1.0D0,  1.0D0,  1.0D0,  9.0D0,  6.0D0,
     &     4.5D0,  1.0D0,  5.0D0,  6.0D0,  8.8D0,
     &     1.0D0,  6.8D0,  1.0D0,  1.0D0,  8.3D0,
     &     7.0D0,  7.0D0,  5.0D0,  4.5D0,  1.0D0/
      DATA (A0(I),I=301,319)
     &    / 4.5D0,  5.0D0,  5.5D0,  1.0D0,  1.0D0,
     &     3.5D0,  0.0D0,  6.0D0,  4.5D0,  0.0D0,
     &     4.0D0,  4.0D0,  4.0D0,  4.0D0,  4.5D0,
     &     4.0D0,  6.0D0,  4.5D0,  0.0D0/
C
      BLANKS='                                        '
      BLANKS=BLANKS//'                                        '
C
C
C     M A I N L I N E
C     ---------------
C
      WRITE(*,*) '    MACS80     PC Version 4.10'
      WRITE(*,*) 'Multi-element Aquatic Chemical Speciation'
      WRITE(*,*)
      CALL INITCM(NKVALS)
      CALL INPUT(PR,NAVALS,NMVALS)
C
C     CALCULATE FOR EACH SPECIES CORRECTION FACTORS FOR THE STABILITY
C     CONSTANT BASED ON IONIC STRENGTH AND TEMPERATURE.
C
      SQRTI=DSQRT(IONIC)
      A1=0.4875*DEXP(1.783D-3*TEMP)
      B1=0.3239*DEXP(5.933D-4*TEMP)
      DO 20 I=1,NGVALS
        IF(Z(I).NE.0)THEN
          GTEMP=(A1*Z(I)**2*SQRTI)/(1D0+B1*A0(I)*SQRTI)
          GAM(I)=1D1**(-GTEMP)
        ELSE
          GAM(I)=1D0
        END IF
   20 CONTINUE
C
      CALL ALPHAC
C
C     CALCULATE K/ALPHA AT THE ASSIGNED TEMPERATURE AND IONIC STRENGTH=
C     KA(34) AND KA(71) ARE COMPUTED SEPARATELY BECAUSE THEY ARE FUNCTI
C     OF IONIC STRENGTH.  KA(12),KA(119),KA(220),KA(282) AND KA(290) AR
C     NOT COMPUTED BECAUSE THE C AND M VALUES ARE NOT DEFINED.  IF THE
C     M(I) VALUE IS 0 THEN KA(I)=DEXP(C(I))/ALPHA(I), THE VALUE AT 25 C
C
      DO 30 I=1,NKVALS
   30 KA(I)=DEXP(C(I)+M(I)/(273.15+TEMP))/ALPHA(I)
      KA(34)=3.12D-3-SQRTI*1.00D-3/ALPHA(34)
      KA(71)=6.40D-4-SQRTI*1.36D-3/ALPHA(71)
      KA(12)=0D0
      KA(119)=0D0
      KA(220)=0D0
      KA(282)=0D0
      KA(290)=0D0
C   UNKNOWN SEDIMENT DISTRIBUTION CONSTANTS
C
C   SEDIMENT COEFFICIENTS ARE A FUNCTION OF PH
```

```
C   K = a * 10**-6 * 10**( k * pH)
      KA(304)=SSCDA
C  SSCd CONSTANT ARBITRARILY MULTIPLIED BY 5 (WAS 1.08613D-5)
C
      KA(305)=0D0
      KA(306)=0D0
      KA(307)=0D0
      KA(308)=0D0
      KA(309)=0D0
      KA(310)=0D0
      KA(311)=0D0
      KA(312)=0D0
      KA(313)=0D0
      KA(314)=0D0
      KA(315)=0D0
      KA(316)=0D0
C
      A(7)=A(7)*6.76D-2
      NUMPH=1
      PH=PHMIN
      DO WHILE(PH.LE.PHMAX)
        WRITE(*,'(''+pH '',F5.2,\)') PH
$IF DEFINED(DEVELOP)
        WRITE(*,'(''    CALCULATING...  pH='',F5.2)') PH
$ENDIF
C
C  INITIALIZATION OF THE VALUES WHICH ARE pH DEPENDENT.
C
        CALL INSPEC(PH,KA,SSCDB)
C
C  ITERATE BETWEEN THE ANION AND METAL SUBSYSTEMS UNTIL EQUILIBRIUM
C  IS REACHED (SUCCESSIVE CONCENTRATIONS OF THE METALS DIFFER BY ONLY
C  A VERY SMALL AMOUNT).
C
        DO 40 I=1,NMVALS
        XMTL(I)=0D0
  40    XPREV(I)=0D0
        DO 45 I=2,NAVALS
  45    XANIN(I)=0D0
        IF(S$PCO2)  XANIN(5)=KA(1)*KA(2)*KA(11)*PCO2*(10D0**PH)**2
        ITER=0
        BETA=1
  50    ITER=ITER+1
        IF(ITER.GT.MAXITR)THEN
          WRITE(PR,60) FF,MAXITR,PH,(I,XMTL(I),I,XPREV(I),I=1,NMVALS)
  60     FORMAT(A/'MAX ITERATIONS(',I3,') EXCEEDED.  pH=',F5.2,
     &    1P,(T43,'XMTL(',I2,')=',D10.4,2X,'XPREV(',I2,')=',D10.4))
          GO TO 110
        END IF
        CALL NWMTLS(NMVALS,NAVALS,PH,MTL,MTLSLD,KA,ITER)
        CALL NWANIN(NMVALS,NAVALS,PH,A,ASLD,KA,CONVTM,S$PCO2,BETA)
```

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  307          IF(ITER.EQ.1)GO TO 50
  308          EQUILB=.TRUE.
  309          DO 100 METAL=1,NMVALS
  310            IF(XPREV(METAL).EQ.0D0)GO TO 100
  311            EQUILB=EQUILB.AND.DABS(1D0-XMTL(METAL)/XPREV(METAL)).LT.TOL
  312     100  CONTINUE
  313          IF (ITER/2*2.EQ.ITER) WRITE(*,'(''*'',\)')
  314          IF(.NOT.EQUILB)GO TO 50
  315  C
  316  C  CALCULATE AND STORE THE CONCENTRATIONS OF ALL AQUEOUS SPECIES
  317  C  AT THIS pH POINT.
  318  C
  319     110   PHS(NUMPH)=PH
  320           CALL SPCALC(PH,NUMPH,NMVALS,NAVALS,BETA,SSCDB)
  321           NUMPH=NUMPH+1
  322           PH=PH+PHINC
  323           WRITE(*,*)
  324           WRITE(*,'(''+'',A60)') BLANKS
  325         END DO
  326         NUMPH=NUMPH-1
  327         CALL OUTPUT(NAVALS,NMVALS,A,MTL)
  328         CLOSE (UNIT=6)
  329         STOP
  330         END


main  Local Symbols

Name                      Class   Type              Size   Offset

EQUILB. . . . . . . . . . local   LOGICAL*2            2    0002
I . . . . . . . . . . . . local   INTEGER*4            4    0004
SQRTI . . . . . . . . . . local   REAL*8               8    0008
PH. . . . . . . . . . . . local   REAL*8               8    0010
BETA. . . . . . . . . . . local   REAL*8               8    0018
A1. . . . . . . . . . . . local   REAL*8               8    0020
B1. . . . . . . . . . . . local   REAL*8               8    0028
METAL . . . . . . . . . . local   INTEGER*4            4    0030
ITER. . . . . . . . . . . local   INTEGER*4            4    0034
BLANKS. . . . . . . . . . local   CHAR*60             60    0038
GTEMP . . . . . . . . . . local   REAL*8               8    0074
NAVALS. . . . . . . . . . local   INTEGER*4            4    0084
NMVALS. . . . . . . . . . local   INTEGER*4            4    0088
NGVALS. . . . . . . . . . local   INTEGER*4            4    008c
NKVALS. . . . . . . . . . local   INTEGER*4            4    0090
MAXITR. . . . . . . . . . local   INTEGER*4            4    0094
TOL . . . . . . . . . . . local   REAL*8               8    0098
FF. . . . . . . . . . . . local   CHAR*1               1    00a0
Z . . . . . . . . . . . . local   INTEGER*2          638    00a2
AO. . . . . . . . . . . . local   REAL*8            2552    0320
TDCU. . . . . . . . . . . OUT1    REAL*4             164    0d74
```

```
main  Local Symbols

Name                          Class   Type              Size   Offset

TDFE2 . . . . . . . . . . . . OUT1    REAL*4             164    0e18
TDMN2 . . . . . . . . . . . . OUT1    REAL*4             164    0ebc
TDZN. . . . . . . . . . . . . OUT1    REAL*4             164    0f60
TDK . . . . . . . . . . . . . OUT1    REAL*4             164    1004
TDCO. . . . . . . . . . . . . OUT1    REAL*4             164    10a8
TDSS. . . . . . . . . . . . . OUT1    REAL*4             164    07b0
TDF . . . . . . . . . . . . . OUT1    REAL*4             164    0854
TDAL. . . . . . . . . . . . . OUT1    REAL*4             164    114c
C . . . . . . . . . . . . . . OUT2    REAL*8            2848    0000
M . . . . . . . . . . . . . . OUT2    REAL*8            2848    0b20
ALPHA . . . . . . . . . . . . OUT2    REAL*8            2848    1640
KA. . . . . . . . . . . . . . OUT2    REAL*8            2848    2160
S$PCO2. . . . . . . . . . . . IN      LOGICAL*2            2    0128
S$DEFNS . . . . . . . . . . . BOTH    LOGICAL*2            2    032c
GAM . . . . . . . . . . . . . OUT2    REAL*8            2552    2c80
KTAG. . . . . . . . . . . . . OUT2    CHAR*2             712    3678
S$ASPEC . . . . . . . . . . . BOTH    LOGICAL*2           30    032e
S$MSPEC . . . . . . . . . . . BOTH    LOGICAL*2           28    034c
AFILES. . . . . . . . . . . . BOTH    CHAR*15            225    0178
UNITS . . . . . . . . . . . . BOTH    INTEGER*4           24    015c
MTLFILES. . . . . . . . . . . BOTH    CHAR*15            210    0259
FILEID. . . . . . . . . . . . BOTH    CHAR*4               4    0174
ASLD. . . . . . . . . . . . . BOTH    INTEGER*4           60    00e8
MTLSLD. . . . . . . . . . . . BOTH    INTEGER*4           56    0124
CONVTA. . . . . . . . . . . . BOTH    REAL*8             120    0000
CONVTM. . . . . . . . . . . . BOTH    REAL*8             112    0078
SPSUM . . . . . . . . . . . . CALC    REAL*8           15120    0000
HSP . . . . . . . . . . . . . CALC    REAL*8             120    3b10
HSPQ. . . . . . . . . . . . . CALC    REAL*8             120    3b88
XANIN . . . . . . . . . . . . CALC    REAL*8             120    3c00
TEMP. . . . . . . . . . . . . IN      REAL*8               8    0000
PCO2. . . . . . . . . . . . . IN      REAL*8               8    0008
XMTL. . . . . . . . . . . . . CALC    REAL*8             112    3c78
IONIC . . . . . . . . . . . . IN      REAL*8               8    0010
XPREV . . . . . . . . . . . . CALC    REAL*8             112    3ce8
CONDUC. . . . . . . . . . . . IN      REAL*8               8    0018
PHMIN . . . . . . . . . . . . IN      REAL*8               8    0020
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020
PHMAX . . . . . . . . . . . . IN      REAL*8               8    0028
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
PHINC . . . . . . . . . . . . IN      REAL*8               8    0030
PHSP. . . . . . . . . . . . . IN      REAL*8               8    0038
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
A . . . . . . . . . . . . . . IN      REAL*8             120    0040
```

```
main  Local Symbols

Name                             Class   Type              Size   Offset

MTL . . . . . . . . . . . . . .  IN      REAL*8             112    00b8
NUMPH . . . . . . . . . . . . .  W       INTEGER*4            4    0018
SSCDA . . . . . . . . . . . . .  IN      REAL*8               8    012a
NPL . . . . . . . . . . . . . .  W       INTEGER*4            4    001c
SSCDB . . . . . . . . . . . . .  IN      REAL*8               8    0132
X . . . . . . . . . . . . . . .  OUT     REAL*8          104632    0000
PHS . . . . . . . . . . . . . .  OUT     REAL*4             164    98b8
TDSO4 . . . . . . . . . . . . .  OUT1    REAL*4             164    0000
TDEDTA. . . . . . . . . . . . .  OUT1    REAL*4             164    00a4
TDNTA . . . . . . . . . . . . .  OUT1    REAL*4             164    0148
TDIC. . . . . . . . . . . . . .  OUT1    REAL*4             164    01ec
TDCL. . . . . . . . . . . . . .  OUT1    REAL*4             164    0290
TDOC. . . . . . . . . . . . . .  OUT1    REAL*4             164    0334
TDSE. . . . . . . . . . . . . .  OUT1    REAL*4             164    03d8
TDSIO4. . . . . . . . . . . . .  OUT1    REAL*4             164    047c
TDNH3 . . . . . . . . . . . . .  OUT1    REAL*4             164    0520
TDPO4 . . . . . . . . . . . . .  OUT1    REAL*4             164    05c4
TDAC. . . . . . . . . . . . . .  OUT1    REAL*4             164    0668
TDS . . . . . . . . . . . . . .  OUT1    REAL*4             164    070c
TDNA. . . . . . . . . . . . . .  OUT1    REAL*4             164    08f8
TDMG. . . . . . . . . . . . . .  OUT1    REAL*4             164    099c
TDCA. . . . . . . . . . . . . .  OUT1    REAL*4             164    0a40
TDFE3 . . . . . . . . . . . . .  OUT1    REAL*4             164    0ae4
TDHG1 . . . . . . . . . . . . .  OUT1    REAL*4             164    0b88
TDHG2 . . . . . . . . . . . . .  OUT1    REAL*4             164    0c2c
TDCD. . . . . . . . . . . . . .  OUT1    REAL*4             164    0cd0

   331  C
   332  C
   333          REAL*8 FUNCTION TERM(P,IC,IQ,IL)
   334  C       --------------------
   335          REAL*8 P,IC,IQ,IL
   336          TERM = P*(P*(P*IC+IQ)+IL)
   337          RETURN
   338          END


TERM  Local Symbols

Name                             Class   Type              Size   Offset

TERM. . . . . . . . . . . . . .  param                             0006
IL. . . . . . . . . . . . . . .  param                             0008
IQ. . . . . . . . . . . . . . .  param                             000c
IC. . . . . . . . . . . . . . .  param                             0010
P . . . . . . . . . . . . . . .  param                             0014
```

```
C
C
      REAL*8 FUNCTION LOGFN(X)
C     ----------------------
      REAL*8 X,DLOG10
      IF (X.EQ.0D0) THEN
        LOGFN=0D0
      ELSE
        LOGFN=DLOG10(X)
      END IF
      RETURN
      END
```

```
LOGFN  Local Symbols

Name                        Class   Type              Size   Offset

LOGFN . . . . . . . . . . . param                             0006
X . . . . . . . . . . . . . param                             0008
```

```
C
C
      SUBROUTINE INITCM(NKVALS)
C     -------------------
C
C     THIS BLOCK INITIALIZES THE C AND M VALUES FOR THE
C     CALCULATION OF THE K VALUES.  THE TRUE VALUE OF C(193)
C     IS 180.0622 NOT 160.  THE MAXIMUM VALUE OF THE
C     ARGUMENT OF THE EXP FUNCTION IS 174.673.
C
      INTEGER NKVALS,I
      REAL*8 C(356),M(356),ALPHA(356),KA(356),GAM(319)
      CHARACTER KTAG(356)*2
      COMMON /OUT2/C,M,ALPHA,KA,GAM,KTAG
C
C     INITIALIZATION OF THE C VALUES.
C
      DATA (C(I),I=1,50)
     &/-8.971D0,   -1.6402D1, -5.30D0,  -7.25D0,   -6.63D0,
     & -2.87422D1, -3.845D1,  -8.825D0, -7.5838D0, -5.18D0,
     & -1.2063D1,   0D0,      -2.99D0,  -5.94D0,    1.220083D1,
     & -6.91D0,    -8.92436D0, 1.550D1,  2.305D1,  -4.686D1,
     & -1.772D1,   -3.150D1,  -6.171D1, -9.118D1,  -2.434D1,
     &  5.41D0,     2.832D1,   0.92D0,   0.26D0,   -4.449D1,
     &  9.31D0,     1.720D1,   2.360D1,  0D0,      -0.1271D0,
     &  5.14D0,     8.36D0,    1.095D1, -2.218D1,  -3.3042D1,
     & -2.326D1,   -4.674D1,  -7.741D1, -1.096D2,  -2.164D1,
     &  5.66D0,     4.56D0,    5.99D0,   5.53D0,    5.87D0/
      DATA (C(I),I=51,100)
     &/ 1.049D1,    1.358D1,   7.39528D0,1.4056079D1,2.843D1,
     &  4.44D0,     7.25D0,    4.99D0,   6.460156D0, 1.9655518D1,
     & -6.217D1,   -3.164D1,  -8.312D1, -1.117D2,   -7.778D1,
     & -1.05827D2, -8.496D1,  -7.507D1, -1.5768D2,  -7.953D1,
     &  0D0,        1.4873D1,  6.310D0,  3.463D0,   -1.34305D2,
     & -1.5151D1,  -6.6593D1, -1.07998D2,1.382D1,    6.701D0,
     &  1.84D0,    -1.1973D2, -1.3403D1,-5.8025D1,  -9.625D1,
     &  0.5347D0,   0.9560D0, -1.79253D1,2.2105D1,   5.066D0,
     &  8.289D0,    8.289D0,   3.224D0, -2.26900D1, -2.67355D1,
     & -8.2783D1,  -3.49542D1, 1.7458D1, 3.426D1,    3.864D1/
      DATA (C(I),I=101,150)
     &/ 2.4407D1,   5.0196D1,  4.8124D1, 2.4637D1,   8.1972D1,
     & -2.0380D1,   8.5660D0,  2.9795D1, 3.2236D1,   1.9895D1,
     &  3.9239D1,   4.1171D1,  1.2134D1, 2.1851D1,   3.1154D1,
     &  8.6891D1,   3.0860D0,  5.5250D0, 0D0,       -2.3162D1,
     & -1.7386D1,  -1.6909D1,  1.63295D1,8.3784D0,   5.0626D0,
     &  1.87428D1,  2.97266D1, 1.18277D1,-3.19069D1,-1.79141D1,
     & -1.23418D1,  7.8794D0,  1.77299D1,7.8288D0,   4.16768D1,
     &  4.1447D0,   1.09093D1, 9.67D-02, 3.2236D0,   2.3026D0,
     &  3.5460D0,   3.9144D0,  1.15129D1,2.55586D1,  3.13152D1,
     &  1.15129D1,  5.526D0,   6.5163D0, 5.4801D0,   5.9867D0/
      DATA (C(I),I=151,200)
     & /3.4539D0,    0.9901D0,  2.87823D0,1.1513D0,   3.6151D0,
     &  2.9289D0,   4.4210D0,  5.0887D0, 1.03616D1,  1.57957D1,
     &  1.05204D1, -0.4605D0,  3.2236D0, 1.6118D0,  -0.4145D0,
     & -1.1513D0,   2.7516D0,  1.9572D0,-1.6118D0,   4.00650D1,
     &  1.19734D1,  1.63023D1, 2.22476D1,5.13476D1,  5.7795D1 ,
     &  1.144385D2, 1.911D1,   7.98997D0,1.98018D1,  1.23879D1,
     &  1.76292D1,  4.9045D0, -1.8703D0, 7.8518D0,   1.50129D1,
     &  1.9572D1,   6.8732D0,  2.0493D1, 2.27773D1,  3.01639D1,
     &  6.02356D1,  1.287145D2,1.60D2,   1.13057D1,  1.6648D0,
     &  8.5426D0,   8.3814D0,  1.0131D0, 6.6545D0,  -8.93403D1/
      DATA (C(I),I=201,250)
     &/-2.35661D1, -2.73209D1,-1.36458D1,-2.03448D1,-1.67398D1,
     & -2.42934D1, -3.02010D1,-2.30259D1,-1.31304D2,-1.86448D1,
     & -1.65786D1, -4.12393D1,-8.93403D1,-9.55573D1,-9.83204D1,
     & -3.69565D1, -6.07882D1,-2.85521D1,-1.41148D1, 0D0,
     &  1.56285D1,  7.1031D0,  9.2647D0,  5.8569D0,  2.1561D0,
     &  2.70070D1,  8.1281D0,  1.40000D1, 6.5844D0,  3.79346D1,
     &  3.7302D0,   0.8189D0,  2.24042D1, 1.96764D1, 1.82625D1,
     &  6.6733D0,   2.17134D1, 2.30179D1, 6.7278D0,  2.99507D1,
     &  4.2117D0,   1.3440D0,  2.68441D1, 3.6361D0,  2.16213D1,
     &  2.33943D1,  2.47781D1, 5.82D0,    3.03190D1, 3.8684D0/
      DATA (C(I),I=251,300)
     &/ 7.6503D0,   2.5789D0,  3.10830D1, 4.3834D0,  9.9011D0,
     &  1.93417D1,  2.23351D1, 2.34864D1, 6.2170D0,  7.7968D0,
     & -0.8183D0,   8.0590D0,  1.02005D1, 5.0196D0,  3.3618D0,
     &  1.58994D1,  7.1129D0,  4.4879D0,  2.26342D1, 1.51151D1,
     &  9.6377D0,   2.18055D1, 3.31802D1, 8.0821D0,  2.70323D1,
     &  2.63541D1,  2.75203D1, 8.2898D0,  2.92052D1, 2.74828D1,
     &  1.24240D1,  0D0,       4.47286D1, 5.85778D1, 1.20655D1,
     &  2.05851D1,  2.56487D1, 2.93498D1, 1.04998D1, 0D0,
     &  1.83937D1,  2.06462D1, 9.3260D0,  3.62427D1, 5.4341D0,
     &  2.21592D1,  2.08930D1,-2.29798D1,-3.43085D1,-7.98997D1/
      DATA (C(I),I=301,350)
     &/-4.90451D1,  2.27035D1, 3.01639D1, 0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,       0D0,
     &  0D0,        2.07431D1, 4.3058D1,  6.21698D1, 7.59852D1,
     &  4.67448D1,  9.69397D1, 3.75345D1, 4.32897D1, 2.39698D1,
     &  2.62487D1,  3.06268D1, 1.34702D1,-6.81240D0, 3.4769D0,
     &  1.94102D1,  7.36520D0, 1.17440D1, 6.90780D0, 1.22061D1,
     &  1.74994D1,  0D0,      -7.71371D1,-7.18171D1,-4.19999D1,
     & -6.80408D1,  7.29910D0, 1.61181D1, 2.90123D1, 3.84528D1,
     &  4.39794D1,  4.1447D0,  2.5329D0,  2.7632D0,  3.6841D0/
      DATA (C(I),I=351,356)
     &/ 2.6479D0,   1.38155D1, 2.41764D1, 2.99336D1,-1.8351D1,
     & -2.39699D1 /
C
C
C     INITIALIZATION OF THE M VALUES.
C
      DATA (M(I),I=1,50)
     &/-1.67985D3, -2.19843D3, 0D0,       0D0,       0D0,
     &  2.8028D3,   0D0,       1.944475D3,1.6284108D3,0D0,
     &  2.58956D3,  0D0,       0D0,       0D0,      -2.27069147D3,
     &  0D0,       -6.9487892D3, 0D0,     0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,      -6.3073040D3,
     &  0D0,        0D0,       0D0,       0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,       0D0,
     &  0D0,        0D0,       0D0,       0D0,       0D0/
      DATA (M(I),I=51,100)
     &/ 0D0,       0D0,       -7.3025329D2,7.5198680D2,0D0,
     &  0D0,       0D0,        0D0,      2.9065976D3,3.68225497D3,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     & -6.679975D3,-1.4420333D4,-1.6617333D4,0D0,    0D0,
     &  0D0,       0D0,        0D0,      -5.2836D3, -2.395166D3,
     &  0D0,       3.089648D3, 0D0,       0D0,       0D0/
      DATA (M(I),I=101,150)
     &/ 0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,      -1.05594D4,
     & -3.09965D4,-3.15498D4,  6.4408D2, -8.503911D2,0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,      -1.51D1,     0D0,       0D0,       0D0,
     &  0D0,      -1.70014D3,  0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0/
      DATA (M(I),I=151,200)
     &/ 0D0,       0D0,        0D0,       0D0,       0D0,
     &  6.036D1,   0D0,        0D0,       0D0,       0D0,
     &  1.1013185D4,0D0,       0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        1.4737900D3,0D0,      0D0,
     &  0D0,       0D0,        0D0,      -3.1286600D3,0D0,
     & -4.2377750D3,0D0,       2.2132D1, 0D0,        0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0/
      DATA (M(I),I=201,250)
     &/-1.7612000D3,1.7612000D3,-3.1400D3,-3.1903D3, 0D0,
     & -3.4117D3, -2.43297D3,  0D0,       1.49144D4,-9.8879D3,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  2.853719D3,2.183828D3,-7.044622D2,-1.509562D2,7.044622D2,
     & -1.7611555D3,0D0,       3.3210360D3,0D0,      6.0382474D3,
     &  0D0,       5.0318728D3,0D0,       2.5159364D3,9.5102396D3,
     &  3.522311D2, 0D0,       4.5790043D3,2.012749D2,4.1261357D3,
     &  1.0063746D3,1.5095618D3,2.0127491D3,1.0063746D3,0D0,
     &  0D0,       2.3146615D3,6.038247D2,2.4656177D3,1.1070120D3/
      DATA (M(I),I=251,300)
     &/ 0D0,       0D0,        2.1133866D3,9.560558D2,0D0,
     &  0D0,       0D0,        0D0,       0D0,      -7.044622D2,
     &  1.6101993D3,0D0,       0D0,       0D0,       0D0,
     &  2.3146615D3,-1.006375D2,-1.006375D2,-2.2140240D3,7.044622D2,
     &  3.9248608D3,0D0,       0D0,       0D0,       0D0,
     &  5.03187D1, 2.4152990D3,5.5350601D3,9.560558D2,4.4783668D3,
     &  0D0,       0D0,       -1.6101993D3,0D0,      0D0,
     &  0D0,       4.528686D2, 1.8114742D3,0D0,      0D0,
     &  2.0127491D3,4.5790043D3,5.5350601D3,0D0,     0D0,
     & -7.044622D2,2.0630679D3,0D0,       0D0,       0D0/
      DATA (M(I),I=301,350)
     &/ 0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0,
     &  0D0,       0D0,        0D0,       0D0,       0D0/
      DATA (M(I),I=351,356)
     &/  0D0,       0D0,       0D0,        0D0,       0D0,
     &   0D0 /
C
```

```
   533          DO 10 I=1,NKVALS
   534            IF(M(I).EQ.0D0)THEN
   535              KTAG(I)='  '
   536            ELSE
   537              KTAG(I)='* '
   538            END IF
   539       10 CONTINUE
   540    C
   541          RETURN
   542          END


INITCM  Local Symbols

Name                          Class   Type              Size   Offset

NKVALS. . . . . . . . . . . . param                            0006
I . . . . . . . . . . . . . . local   INTEGER*4            4   007c
C . . . . . . . . . . . . . . OUT2    REAL*8            2848   0000
M . . . . . . . . . . . . . . OUT2    REAL*8            2848   0b20
ALPHA . . . . . . . . . . . . OUT2    REAL*8            2848   1640
KA. . . . . . . . . . . . . . OUT2    REAL*8            2848   2160
GAM . . . . . . . . . . . . . OUT2    REAL*8            2552   2c80
KTAG. . . . . . . . . . . . . OUT2    CHAR*2             712   3678

   543    C
   544          SUBROUTINE ALPHAC
   545    C     -------------------
   546          IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
   547          REAL*8 C(356),M(356),ALPHA(356),KA(356),GAM(319)
   548          CHARACTER KTAG(356)*2
   549          COMMON /OUT2/C,M,ALPHA,KA,GAM,KTAG
   550    C
   551    C     THIS BLOCK COMPUTES A CORRECTION FOR EACH STABILITY CONSTANT K.
   552    C     GAMMA VARIABLES WITH A VALUE OF 1.0 HAVE EITHER BEEN OMITTED OR
   553    C     REPLACED WITH THE VALUE 1.0.
   554    C
   555          ALPHA( 1) = GAM(2)
   556          ALPHA( 2) = GAM(1)/GAM(2)
   557          ALPHA( 3) = GAM(3)*GAM(6)
   558          ALPHA( 4) = GAM(3)*GAM(1)
   559          ALPHA( 5) = GAM(5)*GAM(1)
   560          ALPHA( 6) = GAM(3)*GAM(1)
   561          ALPHA( 7) = GAM(3)*GAM(5)*GAM(1)**2
   562          ALPHA( 8) = GAM(3)*GAM(2)/GAM(11)
   563          ALPHA( 9) = GAM(5)*GAM(2)/GAM(12)
   564          ALPHA(10) = GAM(5)*GAM(6)
   565          ALPHA(11) = GAM(4)
   566          ALPHA(12) = 1D0
   567          ALPHA(13) = GAM(3)*GAM(17)/GAM(13)
   568          ALPHA(14) = GAM(5)*GAM(17)/GAM(14)
```

```
  569         ALPHA(15) = GAM(15)/GAM(6)
  570         ALPHA(16) = 1D0/GAM(15)
  571         ALPHA(17) = GAM(17)
  572         ALPHA(18) = 1D0/(GAM(18)*GAM(1))
  573         ALPHA(19) = GAM(20)/(GAM(18)*GAM(1)**2)
  574         ALPHA(20) = GAM(18)
  575         ALPHA(21) = GAM(21)/GAM(18)
  576         ALPHA(22) = 1D0/GAM(18)
  577         ALPHA(23) = GAM(23)/GAM(18)
  578         ALPHA(24) = GAM(24)/GAM(18)
  579         ALPHA(25) = GAM(25)/GAM(18)**2
  580         ALPHA(26) = 1D0/(GAM(6)*GAM(18))
  581         ALPHA(27) = GAM(27)/(GAM(18)*GAM(1))
  582         ALPHA(28) = GAM(28)/(GAM(18)*GAM(16))
  583         ALPHA(29) = GAM(29)/(GAM(18)*GAM(16)**2)
  584         ALPHA(30) = GAM(18)*GAM(17)**2
  585         ALPHA(31) = GAM(31)/GAM(18)
  586         ALPHA(32) = GAM(32)/GAM(18)
  587         ALPHA(33) = GAM(33)/GAM(18)
  588         ALPHA(34) = 1D0
  589         ALPHA(35) = 1D0/GAM(34)
  590         ALPHA(36) = GAM(36)/(GAM(18)*GAM(38))
  591         ALPHA(37) = GAM(37)/(GAM(18)*GAM(38)**2)
  592         ALPHA(38) = 1D0/GAM(38)
  593         ALPHA(39) = GAM(18)*GAM(1)
  594         ALPHA(40) = GAM(40)*GAM(17)**2
  595         ALPHA(41) = GAM(41)/GAM(40)
  596         ALPHA(42) = 1D0/GAM(40)
  597         ALPHA(43) = GAM(43)/GAM(40)
  598         ALPHA(44) = GAM(44)/GAM(40)
  599         ALPHA(45) = GAM(45)/GAM(40)**2
  600         ALPHA(46) = GAM(46)/(GAM(40)*GAM(6))
  601         ALPHA(47) = GAM(47)/(GAM(40)*GAM(16))
  602         ALPHA(48) = GAM(48)/(GAM(40)*GAM(16)**2)
  603         ALPHA(49) = GAM(49)/(GAM(40)*GAM(16)**3)
  604         ALPHA(50) = GAM(50)/GAM(40)
  605         ALPHA(51) = GAM(51)/GAM(40)
  606         ALPHA(52) = GAM(52)/GAM(40)
  607         ALPHA(53) = GAM(53)/GAM(54)
  608         ALPHA(54) = GAM(54)/GAM(55)
  609         ALPHA(55) = GAM(55)/GAM(62)
  610         ALPHA(56) = GAM(56)/(GAM(40)*GAM(38))
  611         ALPHA(57) = 1D0/(GAM(40)*GAM(38)**2)
  612         ALPHA(58) = GAM(58)/(GAM(40)*GAM(38)**3)
  613         ALPHA(59) = GAM(59)/GAM(60)
  614         ALPHA(60) = GAM(60)/GAM(61)
  615         ALPHA(61) = GAM(61)*GAM(40)
  616         ALPHA(62) = GAM(40)*GAM(1)
  617         ALPHA(63) = GAM(18)*GAM(61)
  618         ALPHA(64) = GAM(61)*GAM(18)**2
  619         ALPHA(65) = GAM(1)*(GAM(18)*GAM(17))**2
```

```
      ALPHA(66) = (GAM(1)*GAM(17))**2*GAM(18)**3
      ALPHA(67) = GAM(18)**3*GAM(62)**2
      ALPHA(68) = GAM(40)**3*GAM(62)**2
      ALPHA(69) = GAM(18)**4*GAM(6)*GAM(17)**6
      ALPHA(70) = GAM(18)**2*GAM(16)*GAM(17)**3
      ALPHA(71) = 1D0
      ALPHA(72) = GAM(64)/(GAM(3)*GAM(62))
      ALPHA(73) = GAM(65)/(GAM(3)*GAM(55))
      ALPHA(74) = GAM(66)/(GAM(3)*GAM(54))
      ALPHA(75) = GAM(3)**5*GAM(62)**3*GAM(17)
      ALPHA(76) = GAM(3)*GAM(55)
      ALPHA(77) = GAM(3)**3*GAM(62)**2
      ALPHA(78) = GAM(3)**4*GAM(62)**3
      ALPHA(79) = GAM(67)/(GAM(5)*GAM(62))
      ALPHA(80) = GAM(68)/(GAM(5)*GAM(55))
      ALPHA(81) = GAM(69)/(GAM(5)*GAM(54))
      ALPHA(82) = GAM(5)**5*GAM(62)**3*GAM(17)
      ALPHA(83) = GAM(5)*GAM(55)
      ALPHA(84) = GAM(5)**3*GAM(62)**2
      ALPHA(85) = GAM(5)**4*GAM(62)**3
      ALPHA(86) = GAM(73)/GAM(72)
      ALPHA(87) = GAM(74)/GAM(72)
      ALPHA(88) = GAM(75)/GAM(72)
      ALPHA(89) = GAM(76)/(GAM(72)*GAM(17)**4)
      ALPHA(90) = GAM(80)/(GAM(72)*GAM(6))
      ALPHA(91) = GAM(77)/(GAM(72)*GAM(55))
      ALPHA(92) = GAM(78)/(GAM(72)*GAM(54))
      ALPHA(93) = GAM(79)/(GAM(72)*GAM(38))
      ALPHA(94) = GAM(72)*GAM(61)
      ALPHA(95) = GAM(72)*GAM(17)**2
      ALPHA(96) = GAM(72)**3*GAM(62)**2
      ALPHA(97) = GAM(72)*GAM(1)
      ALPHA(98) = GAM(81)/(GAM(40)*GAM(60))
      ALPHA(99) = GAM(82)/(GAM(40)*GAM(60)**2)
      ALPHA(100) = GAM(83)/(GAM(40)*GAM(60)**3)
      ALPHA(101) = GAM(84)/(GAM(103)*GAM(17))
      ALPHA(102) = GAM(85)/(GAM(103)*GAM(17)**2)
      ALPHA(103) = GAM(86)/(GAM(103)*GAM(17)**3)
      ALPHA(104) = GAM(87)/(GAM(103)**2*GAM(17))
      ALPHA(105) = GAM(88)/(GAM(103)*GAM(17))**3
      ALPHA(106) = GAM(89)/(GAM(103)*GAM(17)*GAM(16))
      ALPHA(107) = GAM(90)/(GAM(103)*GAM(16))
      ALPHA(108) = GAM(91)/(GAM(103)*GAM(16)**2)
      ALPHA(109) = GAM(92)/(GAM(103)*GAM(16)**3)
      ALPHA(110) = GAM(93)/(GAM(103)*GAM(35))
      ALPHA(111) = GAM(94)/(GAM(103)*GAM(35)**2)
      ALPHA(112) = GAM(95)/(GAM(103)*GAM(35)**3)
      ALPHA(113) = GAM(96)/(GAM(103)*GAM(38))
      ALPHA(114) = GAM(97)/(GAM(103)*GAM(38)**2)
      ALPHA(115) = GAM(98)/(GAM(103)*GAM(38)**3)
      ALPHA(116) = GAM(99)/(GAM(103)*GAM(60)**2)
      ALPHA(117) = GAM(100)/(GAM(103)*GAM(6))
      ALPHA(118) = GAM(101)/(GAM(103)*GAM(6)**2)
      ALPHA(119) = GAM(102)/(GAM(103)*GAM(71)**2)
      ALPHA(120) = GAM(103)*GAM(17)**2
      ALPHA(121) = GAM(103)*GAM(61)
      ALPHA(122) = GAM(103)*GAM(61)
      ALPHA(123) = GAM(105)/GAM(104)
      ALPHA(124) = 1D0/GAM(105)
      ALPHA(125) = GAM(107)/GAM(105)**2
      ALPHA(126) = GAM(107)/GAM(108)
      ALPHA(127) = GAM(109)/(GAM(103)*GAM(104))
      ALPHA(128) = GAM(110)/(GAM(40)*GAM(104)**2)
      ALPHA(129) = GAM(103)*GAM(104)
      ALPHA(130) = GAM(18)*GAM(104)
      ALPHA(131) = GAM(5)*GAM(104)
      ALPHA(132) = GAM(115)/(GAM(179)*GAM(17))
      ALPHA(133) = GAM(116)/(GAM(179)*GAM(17)**4)
      ALPHA(134) = GAM(111)/(GAM(179)**2*GAM(17))
      ALPHA(135) = GAM(112)/(GAM(179)**2*GAM(17)**3)
      ALPHA(136) = GAM(113)/(GAM(179)*GAM(2))
      ALPHA(137) = GAM(114)/(GAM(179)*GAM(6))
      ALPHA(138) = GAM(117)/(GAM(179)*GAM(16))
      ALPHA(139) = GAM(118)/(GAM(179)*GAM(38))
      ALPHA(140) = GAM(119)/(GAM(179)*GAM(35))
      ALPHA(141) = GAM(120)/(GAM(179)*GAM(35)**2)
      ALPHA(142) = GAM(121)/(GAM(179)*GAM(35)**3)
      ALPHA(143) = GAM(122)/(GAM(180)*GAM(17))
      ALPHA(144) = GAM(123)/(GAM(180)*GAM(17)**2)
      ALPHA(145) = GAM(124)/(GAM(180)*GAM(17)**3)
      ALPHA(146) = GAM(125)/(GAM(180)**2*GAM(17))
      ALPHA(147) = GAM(126)/(GAM(180)*GAM(55))
      ALPHA(148) = GAM(127)/(GAM(180)*GAM(54))
      ALPHA(149) = GAM(128)/(GAM(180)*GAM(6))
      ALPHA(150) = GAM(129)/(GAM(180)*GAM(6)**2)
      ALPHA(151) = GAM(130)/(GAM(180)*GAM(6)**3)
      ALPHA(152) = GAM(131)/(GAM(180)*GAM(16))
      ALPHA(153) = GAM(132)/(GAM(180)*GAM(16)**2)
      ALPHA(154) = GAM(133)/(GAM(180)*GAM(16)**3)
      ALPHA(155) = GAM(134)/(GAM(180)*GAM(38))
      ALPHA(156) = GAM(135)/(GAM(180)*GAM(38)**2)
      ALPHA(157) = GAM(136)/(GAM(180)*GAM(38)**3)
      ALPHA(158) = GAM(137)/(GAM(180)*GAM(35))
      ALPHA(159) = GAM(138)/(GAM(180)*GAM(35)**2)
      ALPHA(160) = GAM(139)/(GAM(180)*GAM(35)**3)
      ALPHA(161) = GAM(140)/(GAM(180)*GAM(60)*GAM(17))
      ALPHA(162) = GAM(141)/(GAM(181)*GAM(17))
      ALPHA(163) = GAM(142)/(GAM(181)*GAM(55))
      ALPHA(164) = GAM(143)/(GAM(181)*GAM(6))
      ALPHA(165) = GAM(144)/(GAM(181)*GAM(38))
      ALPHA(166) = GAM(145)/(GAM(182)*GAM(17))
      ALPHA(167) = GAM(146)/(GAM(182)*GAM(55))
```

```
 722         ALPHA(168) = GAM(147)/(GAM(182)*GAM(6))
 723         ALPHA(169) = GAM(148)/(GAM(182)*GAM(16))
 724         ALPHA(170) = GAM(149)/(GAM(183)*GAM(17))
 725         ALPHA(171) = GAM(150)/(GAM(183)*GAM(6))
 726         ALPHA(172) = GAM(151)/(GAM(183)*GAM(6)**2)
 727         ALPHA(173) = GAM(152)/(GAM(184)*GAM(17))
 728         ALPHA(174) = GAM(153)/(GAM(184)*GAM(17)**2)
 729         ALPHA(175) = GAM(154)/(GAM(184)*GAM(17))**2
 730         ALPHA(176) = GAM(155)/(GAM(184)**3*GAM(17)**4)
 731         ALPHA(177) = GAM(156)/(GAM(184)*GAM(55))
 732         ALPHA(178) = GAM(157)/(GAM(184)*GAM(54))
 733         ALPHA(179) = GAM(158)/(GAM(184)*GAM(6))
 734         ALPHA(180) = GAM(159)/(GAM(184)*GAM(6)**2)
 735         ALPHA(181) = GAM(160)/(GAM(184)*GAM(16))
 736         ALPHA(182) = GAM(161)/(GAM(184)*GAM(16)**2)
 737         ALPHA(183) = GAM(162)/(GAM(184)*GAM(16)**3)
 738         ALPHA(184) = GAM(163)/(GAM(184)*GAM(38))
 739         ALPHA(185) = GAM(164)/(GAM(184)*GAM(38)**2)
 740         ALPHA(186) = GAM(165)/(GAM(184)*GAM(38)**3)
 741         ALPHA(187) = GAM(166)/(GAM(184)*GAM(105))
 742         ALPHA(188) = GAM(167)/(GAM(184)*GAM(169))
 743         ALPHA(189) = GAM(168)/GAM(169)
 744         ALPHA(190) = GAM(169)/GAM(186)
 745         ALPHA(191) = GAM(170)/GAM(186)**2
 746         ALPHA(192) = GAM(171)/GAM(186)**4
 747         ALPHA(193) = GAM(172)/GAM(186)**4
 748         ALPHA(194) = GAM(173)/(GAM(5)*GAM(186))
 749         ALPHA(195) = GAM(174)/(GAM(5)*GAM(169))
 750         ALPHA(196) = GAM(175)/(GAM(5)*GAM(169)**2)
 751         ALPHA(197) = GAM(176)/(GAM(3)*GAM(186))
 752         ALPHA(198) = GAM(177)/(GAM(3)*GAM(169))
 753         ALPHA(199) = GAM(178)/(GAM(3)*GAM(169)**2)
 754         ALPHA(200) = GAM(5)**2*GAM(169)**3
 755         ALPHA(201) = GAM(179)*GAM(17)**2
 756         ALPHA(202) = GAM(179)*GAM(1)
 757         ALPHA(203) = GAM(179)*GAM(61)
 758         ALPHA(204) = GAM(179)*GAM(61)
 759         ALPHA(205) = GAM(179)*GAM(104)
 760         ALPHA(206) = GAM(180)*GAM(17)**2
 761         ALPHA(207) = GAM(180)*GAM(17)
 762         ALPHA(208) = GAM(180)*GAM(1)
 763         ALPHA(209) = GAM(180)**3*GAM(62)**2
 764         ALPHA(210) = GAM(180)*GAM(61)
 765         ALPHA(211) = GAM(3)*GAM(186)
 766         ALPHA(212) = GAM(183)*GAM(16)**2
 767         ALPHA(213) = GAM(184)*GAM(17)**3
 768         ALPHA(214) = GAM(184)*GAM(17)**3
 769         ALPHA(215) = GAM(184)*GAM(17)**3
 770         ALPHA(216) = GAM(183)*GAM(1)
 771         ALPHA(217) = GAM(184)*GAM(62)
 772         ALPHA(218) = GAM(183)*GAM(55)
```

```
      ALPHA(219) = GAM(183)*GAM(6)
      ALPHA(220) = 1D0/(GAM(180)*GAM(60)**2)
      ALPHA(221) = GAM(190)/GAM(189)
      ALPHA(222) = GAM(191)/GAM(190)
      ALPHA(223) = GAM(192)/GAM(191)
      ALPHA(224) = 1D0/GAM(192)
      ALPHA(225) = GAM(193)/(GAM(181)*GAM(189))
      ALPHA(226) = GAM(195)/(GAM(5)*GAM(189))
      ALPHA(227) = GAM(196)/GAM(195)
      ALPHA(228) = GAM(197)/(GAM(3)*GAM(189))
      ALPHA(229) = GAM(198)/GAM(197)
      ALPHA(230) = GAM(199)/(GAM(184)*GAM(189))
      ALPHA(231) = 1D0/GAM(199)
      ALPHA(232) = GAM(199)/GAM(201)
      ALPHA(233) = GAM(201)/GAM(202)
      ALPHA(234) = GAM(199)/GAM(203)
      ALPHA(235) = GAM(204)/(GAM(103)*GAM(189))
      ALPHA(236) = GAM(205)/GAM(204)
      ALPHA(237) = GAM(204)/GAM(206)
      ALPHA(238) = GAM(207)/(GAM(40)*GAM(189))
      ALPHA(239) = GAM(208)/GAM(207)
      ALPHA(240) = GAM(209)/(GAM(18)*GAM(189))
      ALPHA(241) = GAM(210)/GAM(209)
      ALPHA(242) = GAM(211)/(GAM(209)*GAM(17))
      ALPHA(243) = GAM(212)/(GAM(72)*GAM(189))
      ALPHA(244) = GAM(213)/GAM(212)
      ALPHA(245) = GAM(212)/GAM(214)
      ALPHA(246) = GAM(214)/GAM(215)
      ALPHA(247) = GAM(216)/(GAM(179)*GAM(189))
      ALPHA(248) = GAM(217)/GAM(216)
      ALPHA(249) = GAM(218)/(GAM(180)*GAM(189))
      ALPHA(250) = GAM(219)/GAM(218)
      ALPHA(251) = GAM(220)/(GAM(218)*GAM(17))
      ALPHA(252) = GAM(221)/(GAM(182)*GAM(189))
      ALPHA(253) = GAM(224)/(GAM(222)*GAM(189))
      ALPHA(254) = GAM(223)/GAM(224)
      ALPHA(255) = GAM(225)/(GAM(222)*GAM(17))
      ALPHA(256) = 1D0/(GAM(222)*GAM(17)**2)
      ALPHA(257) = GAM(227)/(GAM(222)*GAM(17)**3)
      ALPHA(258) = GAM(228)/(GAM(222)*GAM(17)**4)
      ALPHA(259) = GAM(229)/(GAM(222)**2*GAM(17))
      ALPHA(260) = 1D0/(GAM(222)*GAM(6))
      ALPHA(261) = GAM(231)/(GAM(222)*GAM(35))
      ALPHA(262) = GAM(232)/(GAM(222)*GAM(35)**2)
      ALPHA(263) = GAM(233)/(GAM(222)*GAM(35)**3)
      ALPHA(264) = 1D0/(GAM(222)*GAM(62))
      ALPHA(265) = GAM(235)/(GAM(222)*GAM(38))
      ALPHA(266) = GAM(237)/GAM(236)
      ALPHA(267) = GAM(238)/GAM(237)
      ALPHA(268) = GAM(239)/GAM(238)
      ALPHA(269) = GAM(241)/(GAM(5)*GAM(236))
```

```
 824         ALPHA(270) = GAM(242)/(GAM(3)*GAM(236))
 825         ALPHA(271) = GAM(243)/(GAM(3)*GAM(236)**2)
 826         ALPHA(272) = GAM(255)/(GAM(72)*GAM(236))
 827         ALPHA(273) = GAM(256)/(GAM(72)*GAM(236)**2)
 828         ALPHA(274) = GAM(257)/GAM(25)
 829         ALPHA(275) = GAM(255)/GAM(258)
 830         ALPHA(276) = GAM(264)/(GAM(224)*GAM(236))
 831         ALPHA(277) = GAM(265)/(GAM(222)*GAM(236)**2)
 832         ALPHA(278) = GAM(264)/GAM(265)
 833         ALPHA(279) = GAM(252)/(GAM(18)*GAM(236))
 834         ALPHA(280) = GAM(253)/(GAM(18)*GAM(236)**2)
 835         ALPHA(281) = GAM(254)/GAM(252)
 836         ALPHA(282) = GAM(252)/GAM(254)
 837         ALPHA(283) = 1D0/(GAM(184)*GAM(236))
 838         ALPHA(284) = GAM(245)/(GAM(184)*GAM(236)**2)
 839         ALPHA(285) = 1D0/GAM(246)
 840         ALPHA(286) = GAM(246)/GAM(247)
 841         ALPHA(287) = GAM(261)/(GAM(180)*GAM(236))
 842         ALPHA(288) = GAM(262)/(GAM(180)*GAM(236)**2)
 843         ALPHA(289) = GAM(263)/GAM(261)
 844         ALPHA(290) = GAM(261)/GAM(263)
 845         ALPHA(291) = GAM(249)/(GAM(40)*GAM(236))
 846         ALPHA(292) = GAM(250)/(GAM(40)*GAM(236)**2)
 847         ALPHA(293) = GAM(249)/GAM(251)
 848         ALPHA(294) = GAM(248)/(GAM(103)*GAM(236))
 849         ALPHA(295) = GAM(240)/(GAM(181)*GAM(236))
 850         ALPHA(296) = GAM(259)/(GAM(179)*GAM(236))
 851         ALPHA(297) = GAM(260)/(GAM(179)*GAM(236)**2)
 852         ALPHA(298) = GAM(222)*GAM(1)
 853         ALPHA(299) = GAM(222)*GAM(17)**2
 854         ALPHA(300) = GAM(222)*GAM(62)**2
 855         ALPHA(301) = GAM(222)*GAM(61)
 856         ALPHA(302) = GAM(269)/GAM(268)
 857         ALPHA(303) = GAM(268)/GAM(168)
 858         ALPHA(304) = 1D0
 859         ALPHA(305) = 1D0
 860         ALPHA(306) = 1D0
 861         ALPHA(307) = 1D0
 862         ALPHA(308) = 1D0
 863         ALPHA(309) = 1D0
 864         ALPHA(310) = 1D0
 865         ALPHA(311) = 1D0
 866         ALPHA(312) = 1D0
 867         ALPHA(313) = 1D0
 868         ALPHA(314) = 1D0
 869         ALPHA(315) = 1D0
 870         ALPHA(316) = 1D0
 871         ALPHA(317) = GAM(285)/(GAM(284)*GAM(17))
 872         ALPHA(318) = GAM(286)/(GAM(284)*GAM(17)**2)
 873         ALPHA(319) = GAM(287)/(GAM(284)*GAM(17)**3)
 874         ALPHA(320) = GAM(288)/(GAM(284)*GAM(17)**4)
 875        ALPHA(321) = GAM(289)/(GAM(284)**2*GAM(17)**2)
 876        ALPHA(322) = GAM(290)/(GAM(284)**3*GAM(17)**4)
 877        ALPHA(323) = GAM(291)/(GAM(284)*GAM(189))
 878        ALPHA(324) = GAM(292)/(GAM(284)*GAM(189))
 879        ALPHA(325) = GAM(293)/(GAM(284)*GAM(189))
 880        ALPHA(326) = GAM(294)/(GAM(284)*GAM(236))
 881        ALPHA(327) = GAM(295)/(GAM(284)*GAM(236))
 882        ALPHA(328) = GAM(296)/(GAM(284)*GAM(236))
 883        ALPHA(329) = GAM(297)/(GAM(284)*GAM(236))
 884        ALPHA(330) = GAM(298)/(GAM(284)*GAM(38))
 885        ALPHA(331) = GAM(299)/(GAM(284)*GAM(1))
 886        ALPHA(332) = GAM(300)/(GAM(284)*GAM(6))
 887        ALPHA(333) = GAM(301)/(GAM(284)*GAM(6)**2)
 888        ALPHA(334) = GAM(302)/(GAM(284)*GAM(62))
 889        ALPHA(335) = GAM(303)/(GAM(284)*GAM(62)**2)
 890        ALPHA(336) = GAM(304)/(GAM(284)*GAM(62)**3)
 891        ALPHA(337) = GAM(305)/(GAM(284)*GAM(71)**2)
 892        ALPHA(338) = GAM(285)/(GAM(284)*GAM(17))
 893        ALPHA(339) = GAM(285)/(GAM(284)*GAM(17))
 894        ALPHA(340) = GAM(285)/(GAM(284)*GAM(17))
 895        ALPHA(341) = GAM(285)/(GAM(284)*GAM(17))
 896        ALPHA(342) = 1D0/GAM(306)
 897        ALPHA(343) = GAM(308)/(GAM(284)*GAM(306))
 898        ALPHA(344) = GAM(309)/(GAM(284)*GAM(306)**2)
 899        ALPHA(345) = GAM(310)/(GAM(284)*GAM(306)**3)
 900        ALPHA(346) = GAM(311)/(GAM(284)*GAM(306)**4)
 901        ALPHA(347) = GAM(312)/(GAM(5)*GAM(306))
 902        ALPHA(348) = GAM(313)/(GAM(3)*GAM(306))
 903        ALPHA(349) = GAM(314)/(GAM(18)*GAM(306))
 904        ALPHA(350) = GAM(315)/(GAM(103)*GAM(306))
 905        ALPHA(351) = GAM(316)/(GAM(180)*GAM(306))
 906        ALPHA(352) = GAM(317)/(GAM(184)*GAM(306))
 907        ALPHA(353) = GAM(318)/(GAM(184)*GAM(306)**2)
 908        ALPHA(354) = GAM(319)/(GAM(184)*GAM(306)**2)
 909        ALPHA(355) = GAM(5)*GAM(306)**2
 910        ALPHA(356) = GAM(3)*GAM(306)**2
 911        RETURN
 912        END
```

```
ALPHAC  Local Symbols

Name                           Class   Type              Size   Offset

C . . . . . . . . . . . . . . OUT2    REAL*8            2848    0000
M . . . . . . . . . . . . . . OUT2    REAL*8            2848    0b20
ALPHA . . . . . . . . . . . . OUT2    REAL*8            2848    1640
KA. . . . . . . . . . . . . . OUT2    REAL*8            2848    2160
GAM . . . . . . . . . . . . . OUT2    REAL*8            2552    2c80
KTAG. . . . . . . . . . . . . OUT2    CHAR*2             712    3678
```

```
C
      SUBROUTINE INSPEC(PH,KA,SSCDB)
C     -------------------
      IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
      REAL*8 PH,KA(356),H,H2,H3,H4,SSCDB
C
      REAL*8 SPSUM(15,14,9),HSP(15),HSPQ(15),
     &       XANIN(15),XMTL(14),XPREV(14)
      COMMON /CALC/SPSUM,HSP,HSPQ,XANIN,XMTL,XPREV
C
C
C     THIS BLOCK INITIALIZES THE SPECIES TABLE AND VARIOUS OTHER
C     VALUES FOR EVALUATING THE SYSTEM AT EACH PH.
C
      H  = 10D0**(-PH)
      H2 = H*H
      H3 = H*H2
      H4 = H*H3
C
      XANIN(1) = KA(17)/H
C
C  INITIALIZATION OF THE SPECIES TABLE.
C
C  NOTE: THE HYDROXIDE SPECIES ARE ALREADY MULTIPLIED BY XANIN(1) SO
C        THAT THERE IS NO NEED FOR ANY SPECIAL INITIALIZATION OF THE
C        METALS EQUATIONS.
C
C NOTE:   SPSUM(ANION,METAL,INDEX)
C               INDEX=1 ... 1 ANION, 1 METAL  IN SPECIES
C                     2 ... 2 ANION, 1 METAL
C                     3 ... 3 ANION, 1 METAL
C                     4 ... 1 ANION, 2 METAL
C                     5 ... 2 ANION, 2 METAL
C                     6 ... 3 ANION, 2 METAL
C                     7 ... 1 ANION, 3 METAL
C                     8 ... 2 ANION, 3 METAL
C                     9 ... 3 ANION, 3 METAL
      SPSUM( 1, 1,1) = XANIN(1)*KA(162)
      SPSUM( 1, 2,1) = XANIN(1)/KA(14)
      SPSUM( 1, 3,1) = XANIN(1)/KA(13)
      SPSUM( 1, 4,1) = XANIN(1)*(XANIN(1)*KA(174)+KA(173))
      SPSUM( 1, 4,4) = XANIN(1)**2*KA(175)
      SPSUM( 1, 4,7) = XANIN(1)**4*KA(176)
      SPSUM( 1, 5,1) = XANIN(1)*KA(170)
      SPSUM( 1, 6,1) = XANIN(1)*KA(101) +XANIN(1)**2*KA(102)
     &               + XANIN(1)**3*KA(103)
      SPSUM( 1, 6,4) = XANIN(1)*KA(104)
      SPSUM( 1, 6,7) = XANIN(1)**3*KA(105)
      SPSUM( 1, 7,1) = KA(41)/H +KA(42)/H2 +KA(43)/H3 +KA(44)/H4
      SPSUM( 1, 7,4) = KA(45)/H
      SPSUM( 1, 8,1) = KA(21)/H +KA(22)/H2 +KA(23)/H3 +KA(24)/H4
```

```
 964        SPSUM( 1, 8,4) = KA(25)/H2
 965        SPSUM( 1, 9,1) = KA(86)/H+KA(87)/H2+KA(88)/H3+KA(89)/H2**2
 966        SPSUM( 1,10,1) = XANIN(1)*KA(132) +XANIN(1)**4*KA(133)
 967        SPSUM( 1,10,4) = XANIN(1)*KA(134) +XANIN(1)**3*KA(135)
 968        SPSUM( 1,11,1) = XANIN(1)*KA(143) +XANIN(1)**2*KA(144)
 969       &                + XANIN(1)**3*KA(145)
 970        SPSUM( 1,11,4) = XANIN(1)*KA(146)
 971        SPSUM( 1,12,1) = XANIN(1)*KA(166)
 972        SPSUM( 1,13,1) = XANIN(1)   *KA(255) +XANIN(1)**2*KA(256)
 973       &                + XANIN(1)**3*KA(257) +XANIN(1)**4*KA(258)
 974        SPSUM( 1,13,4) = XANIN(1)*KA(259)
 975        SPSUM( 1,14,1) = XANIN(1)   *KA(317) +XANIN(1)**2*KA(318)
 976       &                + XANIN(1)**3*KA(319) +XANIN(1)**4*KA(320)
 977        SPSUM( 1,14,4) = XANIN(1)**2*KA(321)
 978        SPSUM( 1,14,7) = XANIN(1)**4*KA(322)
 979  C SO4
 980        SPSUM( 2, 1,1) = KA(164)
 981        SPSUM( 2, 2,1) = 1D0/KA(10)
 982        SPSUM( 2, 3,1) = 1D0/KA(3)
 983        SPSUM( 2, 4,1) = KA(179)
 984        SPSUM( 2, 4,2) = KA(180)
 985        SPSUM( 2, 5,1) = KA(171)
 986        SPSUM( 2, 5,2) = KA(172)
 987        SPSUM( 2, 6,1) = KA(117)
 988        SPSUM( 2, 6,2) = KA(118)
 989        SPSUM( 2, 7,1) = KA(46)
 990        SPSUM( 2, 8,1) = KA(26)
 991        SPSUM( 2, 9,1) = KA(90)
 992        SPSUM( 2,10,1) = KA(137)
 993        SPSUM( 2,11,1) = KA(149)
 994        SPSUM( 2,11,2) = KA(150)
 995        SPSUM( 2,11,3) = KA(151)
 996        SPSUM( 2,12,1) = KA(168)
 997        SPSUM( 2,13,1) = KA(260)
 998        SPSUM( 2,14,1) = KA(332)
 999        SPSUM( 2,14,2) = KA(333)
1000  C EDTA
1001        SPSUM( 3, 1,1) = KA(225)
1002        SPSUM( 3, 2,1) = KA(226) +KA(226)*KA(227)*H
1003        SPSUM( 3, 3,1) = KA(228) +KA(228)*KA(229)*H
1004        SPSUM( 3, 4,1) = KA(230) +KA(230)*KA(231)*H
1005       &                + KA(230)/(KA(232)*H) +KA(230)/(KA(232)*KA(233)*H2
1006        SPSUM( 3, 4,5) = KA(230)**2/H2/KA(234)
1007        SPSUM( 3, 6,1) = KA(235) +KA(235)*KA(236)*H +KA(235)/(KA(237)*H)
1008        SPSUM( 3, 7,1) = KA(238) +KA(238)*KA(239)*H
1009        SPSUM( 3, 8,1) = KA(240) +KA(240)*KA(241)*H
1010       &                + KA(240)*KA(242)*XANIN(1)
1011        SPSUM( 3, 9,1) = KA(243) +KA(243)*KA(244)*H
1012       &                + KA(243)/(KA(245)*H) +KA(243)/(KA(245)*KA(246)*H2
1013        SPSUM( 3,10,1) = KA(247) +KA(247)*KA(248)*H
1014        SPSUM( 3,11,1) = KA(249) +KA(249)*KA(250)*H
```

```
1015       &               + KA(249)*KA(251)*XANIN(1)
1016        SPSUM( 3,12,1) = KA(252)
1017        SPSUM( 3,13,1) = KA(253) +KA(253)*KA(254)*H
1018        SPSUM( 3,14,1) = KA(323) +KA(324)*H +KA(325)/H
1019  C NTA
1020        SPSUM( 4, 1,1) = KA(295)
1021        SPSUM( 4, 2,1) = KA(269)
1022        SPSUM( 4, 3,1) = KA(270)
1023        SPSUM( 4, 3,2) = KA(271)
1024        SPSUM( 4, 4,1) = KA(283) +KA(283)/(KA(285)*H)
1025       &               + KA(283)/(KA(285)*KA(286)*H2)
1026        SPSUM( 4, 4,2) = KA(284)
1027        SPSUM( 4, 6,1) = KA(294)
1028        SPSUM( 4, 7,1) = KA(291) +KA(291)/(KA(293)*H)
1029        SPSUM( 4, 7,2) = KA(292)
1030        SPSUM( 4, 8,1) = KA(279) +KA(279)*KA(281)*XANIN(1)
1031        SPSUM( 4, 8,2) = KA(280)
1032        SPSUM( 4, 9,1) = KA(272) +KA(272)*KA(274)*H
1033       &               + KA(272)/(KA(275)*H)
1034        SPSUM( 4, 9,2) = KA(273)
1035        SPSUM( 4,10,1) = KA(296)
1036        SPSUM( 4,10,2) = KA(297)
1037        SPSUM( 4,11,1) = KA(287) +KA(287)*KA(289)*XANIN(1)
1038        SPSUM( 4,11,2) = KA(288)
1039        SPSUM( 4,13,1) = KA(276) +KA(276)/(KA(278)*H)
1040        SPSUM( 4,13,2) = KA(277)
1041        SPSUM( 4,14,1) = KA(326) +KA(327)*H +KA(328)/H +KA(329)/H2
1042  C
1043        SPSUM( 5, 2,1) = 1D0/KA(5) +H/(KA(2)*KA(9))
1044        SPSUM( 5, 3,1) = 1D0/KA(4) +H/(KA(2)*KA(8))
1045        SPSUM( 5, 8,1) = KA(18) +KA(27)*H
1046        SPSUM( 5, 8,2) = KA(19)
1047        SPSUM( 5,10,1) = KA(136)*H/KA(2)
1048        SPSUM( 5,14,1) = KA(331)
1049  C CL
1050        SPSUM( 6, 4,1) = KA(181)
1051        SPSUM( 6, 4,2) = KA(182)
1052        SPSUM( 6, 4,3) = KA(183)
1053        SPSUM( 6, 6,1) = KA(106)*XANIN(1) +KA(107)
1054        SPSUM( 6, 6,2) = KA(108)
1055        SPSUM( 6, 6,3) = KA(109)
1056        SPSUM( 6, 7,1) = KA(47)
1057        SPSUM( 6, 7,2) = KA(48)
1058        SPSUM( 6, 7,3) = KA(49)
1059        SPSUM( 6, 8,1) = KA(28)
1060        SPSUM( 6, 8,2) = KA(29)
1061        SPSUM( 6,10,1) = KA(138)
1062        SPSUM( 6,11,1) = KA(152)
1063        SPSUM( 6,11,2) = KA(153)
1064        SPSUM( 6,11,3) = KA(154)
1065        SPSUM( 6,12,1) = KA(169)
```

```
C OC
      SPSUM( 7, 6,2) = KA(119)
      SPSUM( 7, 7,2) = KA(71)/H2
      SPSUM( 7, 8,2) = KA(34)/H2
      SPSUM( 7,14,2) = KA(337)/H2
C SE
      SPSUM( 8, 4,1) = KA(187)*KA(123)*H
      SPSUM( 8, 6,2) = KA(127)
      SPSUM( 8, 7,2) = KA(128)
C SI04
      SPSUM( 9, 2,1) = KA(194) +KA(195)*KA(190)*H
      SPSUM( 9, 2,2) = KA(196)*KA(190)**2*H2
      SPSUM( 9, 3,1) = KA(197) +KA(198)*KA(190)*H
      SPSUM( 9, 3,2) = KA(199)*KA(190)**2*H2
      SPSUM( 9, 4,1) = KA(188)*KA(190)*H
C NH3
      SPSUM(10, 6,1) = KA(110)
      SPSUM(10, 6,2) = KA(111)
      SPSUM(10, 6,3) = KA(112)
      SPSUM(10, 7,1) = KA(50)
      SPSUM(10, 7,2) = KA(51)
      SPSUM(10, 7,3) = KA(52)
      SPSUM(10, 8,1) = KA(31)
      SPSUM(10, 8,2) = KA(32)
      SPSUM(10, 8,3) = KA(33)
      SPSUM(10,10,1) = KA(140)
      SPSUM(10,10,2) = KA(141)
      SPSUM(10,10,3) = KA(142)
      SPSUM(10,11,1) = KA(158)
      SPSUM(10,11,2) = KA(159)
      SPSUM(10,11,3) = KA(160)
      SPSUM(10,13,1) = KA(261)
      SPSUM(10,13,2) = KA(262)
      SPSUM(10,13,3) = KA(263)
C PO4
      SPSUM(11, 1,1) = KA(163)*KA(55)*H
      SPSUM(11, 2,1) = KA(79) +KA(55)*KA(80)*H +KA(81)*KA(55)*KA(54)*H2
      SPSUM(11, 3,1) = KA(72) +KA(55)*KA(73)*H +KA(74)*KA(55)*KA(54)*H2
      SPSUM(11, 4,1) = KA(177)*KA(55)*H +KA(178)*KA(55)*KA(54)*H2
      SPSUM(11, 9,1) = KA(91) *KA(55)*H +KA(92) *KA(55)*KA(54)*H2
      SPSUM(11,11,1) = KA(147)*KA(55)*H +KA(148)*KA(55)*KA(54)*H2
      SPSUM(11,12,1) = KA(167)*KA(55)*H
      SPSUM(11,13,1) = KA(264)*KA(55)*H
      SPSUM(11,14,1) = KA(334)*KA(54)*KA(55)*H2
      SPSUM(11,14,2) = KA(335)*KA(54)**2*KA(55)**2*H2**2
      SPSUM(11,14,3) = KA(336)*KA(54)**3*KA(55)**3*H2**3
C AC
      SPSUM(12, 1,1) = KA(165)
      SPSUM(12, 4,1) = KA(184)
      SPSUM(12, 4,2) = KA(185)
      SPSUM(12, 4,3) = KA(186)
      SPSUM(12, 6,1) = KA(113)
      SPSUM(12, 6,2) = KA(114)
      SPSUM(12, 6,3) = KA(115)
      SPSUM(12, 7,1) = KA(56)
      SPSUM(12, 7,2) = KA(57)
      SPSUM(12, 7,3) = KA(58)
      SPSUM(12, 8,1) = KA(36)
      SPSUM(12, 8,2) = KA(37)
      SPSUM(12, 9,1) = KA(93)
      SPSUM(12,10,1) = KA(139)
      SPSUM(12,11,1) = KA(155)
      SPSUM(12,11,2) = KA(156)
      SPSUM(12,11,3) = KA(157)
      SPSUM(12,13,1) = KA(265)
      SPSUM(12,14,1) = KA(330)
C S
      SPSUM(13, 6,2) = KA(116)*KA(60)**2*H2
      SPSUM(13, 7,1) = KA(98)*KA(60)*H
      SPSUM(13, 7,2) = KA(99)*KA(60)**2*H2
      SPSUM(13, 7,3) = KA(100)*KA(60)**3*H3
      SPSUM(13,11,1) = KA(161)*KA(60)*H*XANIN(1)
C SS
      SPSUM(14, 1,1) = KA(312)
      SPSUM(14, 2,1) = KA(313)
      SPSUM(14, 3,1) = KA(314)
      SPSUM(14, 4,1) = KA(307)
      SPSUM(14, 5,1) = KA(315)
      SPSUM(14, 6,1) = KA(316)
      SPSUM(14, 7,1) = KA(304)*10**(SSCDB*PH)
      SPSUM(14, 8,1) = KA(305)
      SPSUM(14, 9,1) = KA(306)
      SPSUM(14,10,1) = KA(308)
      SPSUM(14,11,1) = KA(309)
      SPSUM(14,12,1) = KA(310)
      SPSUM(14,13,1) = KA(311)
C
      SPSUM(15, 2,1) = KA(347)
      SPSUM(15, 3,1) = KA(348)
      SPSUM(15, 4,1) = KA(352)
      SPSUM(15, 4,2) = KA(353)
      SPSUM(15, 4,3) = KA(354)
      SPSUM(15, 6,1) = KA(350)
      SPSUM(15, 8,1) = KA(349)
      SPSUM(15,11,1) = KA(351)
      SPSUM(15,14,1) = KA(343)
      SPSUM(15,14,2) = KA(344)
      SPSUM(15,14,3) = KA(345)
C
C  CONSTANTS FOR SPECIES WITH ONLY ANIONS (I.E. NO METALS).
C
      HSP( 2) = KA(15)*H +KA(15)*KA(16)*H2
```

```
 1168         HSP( 3) = KA(221)*H +KA(221)*(KA(222)*H2
 1169        &         + KA(222)*(KA(223)*H3+KA(223)*KA(224)*H4))
 1170         HSP( 4) = KA(266)*H +KA(266)*(KA(267)*H2+KA(267)*KA(268)*H3)
 1171         HSP( 5) = H/KA(2) +H2/(KA(1)*KA(2))
 1172         HSP( 7) = 10D0**((PH-4.86D0)/1.90D0)
 1173         HSP( 8) = KA(123)*H +KA(123)*KA(124)*H2
 1174         HSPQ(8) = KA(123)**2*(KA(125)*H/(KA(126))+KA(125)*H2)
 1175         HSP( 9) = KA(190)*(H+KA(189)*(H2+KA(303)*(H3+KA(302)*H4)))
 1176         HSPQ(9) = KA(191)*H2
 1177         HSP(10) = H/KA(35)
 1178         HSP(11) = KA(55)*H+KA(55)*(KA(54)*H2+KA(53)*KA(54)*H3)
 1179         HSP(12) = KA(38)*H
 1180         HSP(13) = KA(60)*H +KA(59)*KA(60)*H2
 1181   C       HSP(14) =
 1182         HSP(15) = KA(342)*H
 1183   C
 1184         RETURN
 1185         END


INSPEC  Local Symbols

Name                          Class   Type              Size   Offset

SSCDB . . . . . . . . . . . . param                             0006
KA. . . . . . . . . . . . . . param                             000a
PH. . . . . . . . . . . . . . param                             000e
H . . . . . . . . . . . . . . local   REAL*8               8    0080
H2. . . . . . . . . . . . . . local   REAL*8               8    0088
H3. . . . . . . . . . . . . . local   REAL*8               8    0090
H4. . . . . . . . . . . . . . local   REAL*8               8    0098
SPSUM . . . . . . . . . . . . CALC    REAL*8           15120    0000
HSP . . . . . . . . . . . . . CALC    REAL*8             120    3b10
HSPQ. . . . . . . . . . . . . CALC    REAL*8             120    3b88
XANIN . . . . . . . . . . . . CALC    REAL*8             120    3c00
XMTL. . . . . . . . . . . . . CALC    REAL*8             112    3c78
XPREV . . . . . . . . . . . . CALC    REAL*8             112    3ce8

 1186   C
 1187         SUBROUTINE INPUT(PR,NAVALS,NMVALS)
 1188   C     ----------------
 1189   C
 1190   C     THIS ROUTINE READS THE INPUT DATA AND PRINTS A LISTING OF THE DAT
 1191   C
 1192         REAL*8 CONVTA(15),CONVTM(14),TEMP,PCO2,IONIC,CONDUC,
 1193        &       PHMIN,PHMAX,PHINC,PHSP,AMOUNT,A(15),MTL(14),SSCDA,SSCDB
 1194         INTEGER SOLID,ASLD(15),MTLSLD(14),I,IND,PR,UNITS(6),POS
 1195         LOGICAL*2 S$PCO2,S$INP(6),S$DEFNS
 1196         LOGICAL*2 S$TABLE,S$ASPEC(15),S$MSPEC(14)
 1197         CHARACTER*15 AFILES(15),MTLFILES(14),FILENAME
 1198         CHARACTER FILEID*4,FILEID2*4,MAINFILE*8
```

```
      CHARACTER TITLE*48,UNIT2*2,TYPE*1
      COMMON /IN/TEMP,PCO2,IONIC,CONDUC,
     &   PHMIN,PHMAX,PHINC,PHSP,A,MTL,S$PCO2,SSCDA,SSCDB
      COMMON /BOTH/CONVTA,CONVTM,ASLD,MTLSLD,UNITS,
     &             FILEID,AFILES,MTLFILES,S$DEFNS,S$ASPEC,S$MSPEC
C
      DO 1 I=1,NAVALS
      A(I)=0D0
      ASLD(I)=0
      AFILES(I)='               '
    1 S$ASPEC(I)=.FALSE.
      DO 2 I=1,NMVALS
      MTL(I)=0D0
      MTLSLD(I)=0
      MTLFILES(I)='               '
    2 S$MSPEC(I)=.FALSE.
C
      OPEN (UNIT=5,FILE='MACS80.INP',STATUS='OLD')
      READ(5,10) TITLE,TEMP,PCO2,S$PCO2,IONIC,CONDUC
        IF(IONIC.EQ.0D0)IONIC=CONDUC*1.8D-5
      READ(5,12) PHMIN,PHMAX,PHINC,PHSP
      READ(5,20) SSCDA,SSCDB
      READ(5,14) S$INP(1),S$INP(4),S$INP(2),S$INP(5),S$INP(3),S$INP(6)
      J=1
      DO 5 I=1,6
        IF (S$INP(I)) THEN
          UNITS(J)=I
          J=J+1
        END IF
    5 CONTINUE
      READ(5,15) S$DEFNS
      READ(5,17) FILEID2,FILEID
C
      POS=INDEX(FILEID2,' ')
      IF (POS .EQ. 0) THEN
        MAINFILE=FILEID2//FILEID
      ELSE IF (POS .NE. 1) THEN
        MAINFILE=FILEID2(1:POS-1)//FILEID
      ELSE
        MAINFILE=FILEID
      END IF
C
    3 READ(5,16,END=4) TYPE,IND,AMOUNT,UNIT2,SOLID,S$TABLE,FILENAME
      IF(TYPE.EQ.'A')THEN
        A(IND)=AMOUNT
        IF(UNIT2.EQ.'MG')A(IND)=AMOUNT/CONVTA(IND)
        ASLD(IND)=SOLID
        S$ASPEC(IND)=S$TABLE
        IF (FILEID.NE.'    ') THEN
          POS=INDEX(FILENAME,' ')
          IF (POS.NE.1) THEN
```

```
Line#  Source Line
 1250             AFILES(IND)=FILENAME(1:POS-1)//FILEID
 1251           END IF
 1252         END IF
 1253       END IF
 1254  C
 1255       IF(TYPE.EQ.'M')THEN
 1256         MTL(IND)=AMOUNT
 1257         IF(UNIT2.EQ.'MG')MTL(IND)=AMOUNT/CONVTM(IND)
 1258         MTLSLD(IND)=SOLID
 1259         S$MSPEC(IND)=S$TABLE
 1260         IF (FILEID.NE.'    ') THEN
 1261           POS=INDEX(FILENAME,' ')
 1262           IF (POS.NE.1) THEN
 1263             MTLFILES(IND)=FILENAME(1:POS-1)//FILEID
 1264           END IF
 1265         END IF
 1266       END IF
 1267  C
 1268       GO TO 3
 1269  C
 1270  C  WRITE THE PROGRAM PARAMETERS REPORT.
 1271  C
 1272  C
 1273     4 OPEN (UNIT=6,FILE=MAINFILE//'.DAT',STATUS='UNKNOWN')
 1274       CALL WPARMS(PR,TITLE,MAINFILE)
 1275       CLOSE (UNIT=5,STATUS='KEEP')
 1276  C
 1277    10 FORMAT(T25,A48/T25,F4.1/T25,F7.6,T52,L1/(T25,F10.0))
 1278    12 FORMAT(T20,3(5X,F6.0)/T25,F6.0)
 1279    14 FORMAT(3(/,T25,L1,6X,L1))
 1280    15 FORMAT(T25,L1)
 1281    16 FORMAT(A1,T21,I2,T25,F10.0,A2,T46,I2,T55,L1,T61,A10)
 1282    17 FORMAT(T18,A4,T31,A4)
 1283    20 FORMAT(T23,D13.7,T41,D10.7)
 1284       RETURN
 1285       END


INPUT  Local Symbols

Name                         Class   Type              Size   Offset

NMVALS. . . . . . . . . . .  param                            0006
NAVALS. . . . . . . . . . .  param                            000a
PR. . . . . . . . . . . . .  param                            000e
FILENAME. . . . . . . . . .  local   CHAR*15             15   00a0
TYPE. . . . . . . . . . . .  local   CHAR*1               1   00af
TITLE . . . . . . . . . . .  local   CHAR*48             48   00b0
MAINFILE. . . . . . . . . .  local   CHAR*8               8   00e0
I . . . . . . . . . . . . .  local   INTEGER*4            4   00e8
J . . . . . . . . . . . . .  local   INTEGER*4            4   00ec
```

```
INPUT  Local Symbols

Name                          Class   Type             Size   Offset

AMOUNT. . . . . . . . . . . . local   REAL*8              8    00f0
IND . . . . . . . . . . . . . local   INTEGER*4           4    00f8
S$INP . . . . . . . . . . . . local   LOGICAL*2          12    00fc
FILEID2 . . . . . . . . . . . local   CHAR*4              4    0108
S$TABLE . . . . . . . . . . . local   LOGICAL*2           2    010c
UNIT2 . . . . . . . . . . . . local   CHAR*2              2    010e
POS . . . . . . . . . . . . . local   INTEGER*4           4    0110
SOLID . . . . . . . . . . . . local   INTEGER*4           4    0114
CONVTA. . . . . . . . . . . . BOTH    REAL*8            120    0000
CONVTM. . . . . . . . . . . . BOTH    REAL*8            112    0078
TEMP. . . . . . . . . . . . . IN      REAL*8              8    0000
PCO2. . . . . . . . . . . . . IN      REAL*8              8    0008
IONIC . . . . . . . . . . . . IN      REAL*8              8    0010
CONDUC. . . . . . . . . . . . IN      REAL*8              8    0018
PHMIN . . . . . . . . . . . . IN      REAL*8              8    0020
PHMAX . . . . . . . . . . . . IN      REAL*8              8    0028
PHINC . . . . . . . . . . . . IN      REAL*8              8    0030
PHSP. . . . . . . . . . . . . IN      REAL*8              8    0038
A . . . . . . . . . . . . . . IN      REAL*8            120    0040
MTL . . . . . . . . . . . . . IN      REAL*8            112    00b8
SSCDA . . . . . . . . . . . . IN      REAL*8              8    012a
SSCDB . . . . . . . . . . . . IN      REAL*8              8    0132
ASLD. . . . . . . . . . . . . BOTH    INTEGER*4          60    00e8
MTLSLD. . . . . . . . . . . . BOTH    INTEGER*4          56    0124
UNITS . . . . . . . . . . . . BOTH    INTEGER*4          24    015c
S$PCO2. . . . . . . . . . . . IN      LOGICAL*2           2    0128
S$DEFNS . . . . . . . . . . . BOTH    LOGICAL*2           2    032c
S$ASPEC . . . . . . . . . . . BOTH    LOGICAL*2          30    032e
S$MSPEC . . . . . . . . . . . BOTH    LOGICAL*2          28    034c
AFILES. . . . . . . . . . . . BOTH    CHAR*15           225    0178
MTLFILES. . . . . . . . . . . BOTH    CHAR*15           210    0259
FILEID. . . . . . . . . . . . BOTH    CHAR*4              4    0174

  1286  C
  1287        SUBROUTINE WPARMS(PR,TITLE,MAINFILE)
  1288  C     ------------------
  1289  C
  1290  C     THIS BLOCK WRITES A REPORT CONTAINING ALL INPUT AND
  1291  C     DEFAULT PARAMETERS THAT WILL BE USED FOR THE ANALYSIS.
  1292  C
  1293  C
  1294        IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
  1295        REAL*8 CONVTA(15),CONVTM(14),TEMP,PCO2,IONIC,CONDUC,
  1296       &       PHMIN,PHMAX,PHINC,PHSP,A(15),MTL(14),SSCDA,SSCDB
  1297        INTEGER ASLD(15),MTLSLD(14),PR,UNITS(6)
  1298        LOGICAL*2 S$PCO2,S$DEFNS
```

```
      LOGICAL*2 S$ASPEC(15),S$MSPEC(14)
      CHARACTER*15 AFILES(15),MTLFILES(14)
      CHARACTER FILEID*4,MAINFILE*8
      CHARACTER*48 TITLE
      COMMON /IN/TEMP,PCO2,IONIC,CONDUC,
     &   PHMIN,PHMAX,PHINC,PHSP,A,MTL,S$PCO2,SSCDA,SSCDB
      COMMON /BOTH/CONVTA,CONVTM,ASLD,MTLSLD,UNITS,
     &             FILEID,AFILES,MTLFILES,S$DEFNS,S$ASPEC,S$MSPEC
C
      CHARACTER*12 UNITSLABEL
      CALL PPS(TITLE)
      WRITE(PR,210) TEMP,PCO2,S$PCO2,IONIC,CONDUC
      WRITE(PR,220) PHMIN,PHMAX,PHINC
C
      IF(PHSP.NE.0D0)THEN
        WRITE(PR,230) PHSP
      ELSE
        WRITE(PR,232)
      END IF
      WRITE(PR,300) SSCDA,' *10**(',SSCDB,'*PH)'
      WRITE(PR,240)
      J=1
      DO WHILE (UNITS(J).NE.0 .AND. J.LE.6)
        IF (UNITS(J).EQ.1) UNITSLABEL='MOLES/L     '
        IF (UNITS(J).EQ.2) UNITSLABEL='LOG(MOLES/L)'
        IF (UNITS(J).EQ.3) UNITSLABEL='%(MOLES/L)  '
        IF (UNITS(J).EQ.4) UNITSLABEL='MG/L        '
        IF (UNITS(J).EQ.5) UNITSLABEL='LOG(MG/L)   '
        IF (UNITS(J).EQ.6) UNITSLABEL='%(MG/L)     '
        WRITE(PR,241) UNITSLABEL
        J=J+1
      END DO
      WRITE(PR,250) S$DEFNS
      WRITE(PR,255) MAINFILE
      CALL WFOOT(26-J)
C
C  WRITE OUT THE TOTAL DISSOLVED VALUES, CONTROLLING SOLIDS,
C  AND REQUESTS FOR TABLES.
C
      CALL WHEAD
      WRITE(PR,260)
      WRITE(PR,280) ' '
      WRITE(PR,270) 'SULFATE .....',
     & A( 2),A( 2)*CONVTA( 2),ASLD( 2),S$ASPEC( 2),AFILES( 2)
      WRITE(PR,270) 'EDTA ......',
     & A( 3),A( 3)*CONVTA( 3),ASLD( 3),S$ASPEC( 3),AFILES( 3)
      WRITE(PR,270) 'NTA .......',
     & A( 4),A( 4)*CONVTA( 4),ASLD( 4),S$ASPEC( 4),AFILES( 4)
      WRITE(PR,270) 'INORGANIC CARBON ..',
     & A( 5),A( 5)*CONVTA( 5),ASLD( 5),S$ASPEC( 5),AFILES( 5)
      WRITE(PR,270) 'CHLORIDE ....',
      & A( 6),A( 6)*CONVTA( 6),ASLD( 6),S$ASPEC (6),AFILES( 6)
       WRITE(PR,270) 'ORGANIC CARBON ..',
      & A( 7),A( 7)*CONVTA( 7),ASLD( 7),S$ASPEC( 7),AFILES( 7)
       WRITE(PR,270) 'SELENITE ....',
      & A( 8),A( 8)*CONVTA( 8),ASLD( 8),S$ASPEC( 8),AFILES( 8)
       WRITE(PR,270) 'SILICATE ....',
      & A( 9),A( 9)*CONVTA( 9),ASLD( 9),S$ASPEC( 9),AFILES( 9)
       WRITE(PR,270) 'AMMONIA .....',
      & A(10),A(10)*CONVTA(10),ASLD(10),S$ASPEC(10),AFILES(10)
       WRITE(PR,270) 'ORTHO-PHOSPHATE ...',
      & A(11),A(11)*CONVTA(11),ASLD(11),S$ASPEC(11),AFILES(11)
       WRITE(PR,270) 'ACETATE .....',
      & A(12),A(12)*CONVTA(12),ASLD(12),S$ASPEC(12),AFILES(12)
       WRITE(PR,270) 'SULFIDE .....',
      & A(13),A(13)*CONVTA(13),ASLD(13),S$ASPEC(13),AFILES(13)
       WRITE(PR,271) 'SEDIMENT ....',
      & A(14),                 ASLD(14),S$ASPEC(14),AFILES(14)
       WRITE(PR,270) 'FLUORIDE ....',
      & A(15),A(15)*CONVTA(15),ASLD(15),S$ASPEC(15),AFILES(15)
       WRITE(PR,280) ' '
       WRITE(PR,270) 'SODIUM ......',
      & MTL( 1),MTL( 1)*CONVTM( 1),MTLSLD( 1),S$MSPEC( 1),MTLFILES( 1)
       WRITE(PR,270) 'MAGNESIUM .....',
      & MTL( 2),MTL( 2)*CONVTM( 2),MTLSLD( 2),S$MSPEC( 2),MTLFILES( 2)
       WRITE(PR,270) 'CALCIUM .....',
      & MTL( 3),MTL( 3)*CONVTM( 3),MTLSLD( 3),S$MSPEC( 3),MTLFILES( 3)
       WRITE(PR,270) 'IRON(III) .....',
      & MTL( 4),MTL( 4)*CONVTM( 4),MTLSLD( 4),S$MSPEC( 4),MTLFILES( 4)
       WRITE(PR,270) 'MERCURY(I) ....',
      & MTL( 5),MTL( 5)*CONVTM( 5),MTLSLD( 5),S$MSPEC( 5),MTLFILES( 5)
       WRITE(PR,270) 'MERCURY(II) ...',
      & MTL( 6),MTL( 6)*CONVTM( 6),MTLSLD( 6),S$MSPEC( 6),MTLFILES( 6)
       WRITE(PR,270) 'CADMIUM .....',
      & MTL( 7),MTL( 7)*CONVTM( 7),MTLSLD( 7),S$MSPEC( 7),MTLFILES( 7)
       WRITE(PR,270) 'COPPER ......',
      & MTL( 8),MTL( 8)*CONVTM( 8),MTLSLD( 8),S$MSPEC( 8),MTLFILES( 8)
       WRITE(PR,270) 'IRON(II) ....',
      & MTL( 9),MTL( 9)*CONVTM( 9),MTLSLD( 9),S$MSPEC( 9),MTLFILES( 9)
       WRITE(PR,270) 'MANGANESE(II) ...',
      & MTL(10),MTL(10)*CONVTM(10),MTLSLD(10),S$MSPEC(10),MTLFILES(10)
       WRITE(PR,270) 'ZINC ......',
      & MTL(11),MTL(11)*CONVTM(11),MTLSLD(11),S$MSPEC(11),MTLFILES(11)
       WRITE(PR,270) 'POTASSIUM .....',
      & MTL(12),MTL(12)*CONVTM(12),MTLSLD(12),S$MSPEC(12),MTLFILES(12)
       WRITE(PR,270) 'COBALT ......',
      & MTL(13),MTL(13)*CONVTM(13),MTLSLD(13),S$MSPEC(13),MTLFILES(13)
       WRITE(PR,270) 'ALUMINUM ....',
      & MTL(14),MTL(14)*CONVTM(14),MTLSLD(14),S$MSPEC(14),MTLFILES(14)
       CALL WFOOT(16)
C
  210 FORMAT(' '///T40,'TEMPERATURE .....',T65,F10.6,' DEG C'/
Line#  Source Line        Microsoft FORTRAN Optimizing Compiler Version 5.00

 1401         &/T40,'PCO2 ........',T65,F10.6,' ATM.',
 1402         & T86,'CONTROLS T.D. IC =',L2/
 1403         &/T40,'IONIC STRENGTH ....',T65,F10.6,' MOLES/L'/
 1404         &/T40,'CONDUCTIVITY ....',T65,F10.6,' UMHOS/CM')
 1405     220 FORMAT(/T40,'pH RANGE:'/
 1406       &         T44,'MINIMUM .....',T65,F6.2/
 1407       &         T44,'MAXIMUM .....',T65,F6.2/
 1408       &         T44,'INCREMENT .....',T65,F6.2)
 1409     230 FORMAT(/T40,'pH SPECIFIC .....',T65, F6.2 )
 1410     232 FORMAT(/T40,'pH SPECIFIC .....',T67,'NONE')
 1411     240 FORMAT(/T40,'OUTPUT OPTIONS:'//T44,'UNITS:')
 1412     241 FORMAT(T48,A12)
 1413     250 FORMAT(/T44,'TABLES:'/
 1414       &         T48,'DEFINITIONS ...',T65,L2)
 1415     255 FORMAT(/T44,'OUTPUT FILE:  ',A8,'.DAT')
 1416     260 FORMAT(/T38,'TOTAL DISSOLVED VALUES:',T87,'SOLID',4X,'SPECIES',
 1417       &         6X,'ASCII'/,T63,'MOLES/L',T77,'MG/L',T87,'NUMBER',3X,
 1418       &         'REPORT',5X,'FILE NAME')
 1419     270 FORMAT(T38,A21,T59,1P,2D13.5,T89,I2,8X,L1,8X,A10)
 1420     271 FORMAT(T38,A21,T72,1P, D13.5,T89,I2,8X,L1,8X,A10)
 1421     280 FORMAT(A1)
 1422     300 FORMAT(/,T40,'SEDIMENT CD CONSTANT     ',D13.7,A7,F10.7,A4)
 1423  C
 1424         RETURN
 1425         END


WPARMS  Local Symbols

Name                      Class   Type              Size   Offset

MAINFILE. . . . . . . . . param                             0006
TITLE . . . . . . . . . . param                             000a
PR. . . . . . . . . . . . param                             000e
J . . . . . . . . . . . . local   INTEGER*4            4    0118
UNITSLABEL. . . . . . . . local   CHAR*12             12    011c
CONVTA. . . . . . . . . . BOTH    REAL*8             120    0000
CONVTM. . . . . . . . . . BOTH    REAL*8             112    0078
TEMP. . . . . . . . . . . IN      REAL*8               8    0000
PCO2. . . . . . . . . . . IN      REAL*8               8    0008
IONIC . . . . . . . . . . IN      REAL*8               8    0010
CONDUC. . . . . . . . . . IN      REAL*8               8    0018
PHMIN . . . . . . . . . . IN      REAL*8               8    0020
PHMAX . . . . . . . . . . IN      REAL*8               8    0028
PHINC . . . . . . . . . . IN      REAL*8               8    0030
PHSP. . . . . . . . . . . IN      REAL*8               8    0038
A . . . . . . . . . . . . IN      REAL*8             120    0040
MTL . . . . . . . . . . . IN      REAL*8             112    00b8
SSCDA . . . . . . . . . . IN      REAL*8               8    012a
SSCDB . . . . . . . . . . IN      REAL*8               8    0132
ASLD. . . . . . . . . . . BOTH    INTEGER*4           60    00e8
```

```
WPARMS  Local Symbols

Name                          Class   Type              Size   Offset

MTLSLD. . . . . . . . . . . . BOTH    INTEGER*4           56    0124
UNITS . . . . . . . . . . . . BOTH    INTEGER*4           24    015c
S$PCO2. . . . . . . . . . . . IN      LOGICAL*2            2    0128
S$DEFNS . . . . . . . . . . . BOTH    LOGICAL*2            2    032c
S$ASPEC . . . . . . . . . . . BOTH    LOGICAL*2           30    032e
S$MSPEC . . . . . . . . . . . BOTH    LOGICAL*2           28    034c
AFILES. . . . . . . . . . . . BOTH    CHAR*15            225    0178
MTLFILES. . . . . . . . . . . BOTH    CHAR*15            210    0259
FILEID. . . . . . . . . . . . BOTH    CHAR*4               4    0174
```

```
C
      SUBROUTINE ROOTS(COEFF,R)
C     -------------------------
C
C        "COEFF" IS AN INPUT ARRAY OF THE COEFFICIENTS FOR THE CUBIC
C     EQUATION AND "R" IS A POSITIVE ROOT RETURNED TO THE CALLING
C     PROGRAM.  "COEFF" IS ORDERED BEGINNING WITH THE TERM OF
C     HIGHEST DEGREE.
C
C        THE SOLUTION FOR THE LINEAR CASE AND MOST TIMES THE QUADRATIC
C     CASE IS EXACT.  THE SOLUTION TO THE CUBIC CASE IS OBTAINED USING
C     AN INTERVAL-HALVING SEARCH TECHNIQUE.  AT PRESENT THIS TECHNIQUE
C     IS ACCURATE TO 1 PART IN 1E-4.
C
      REAL*8 A,B,C,D,DISC,DSQRT
      REAL*8 F,R,X,Y1,Y2,Z,COEFF(4),DABS
CCC      REAL*8 DLOG10
      INTEGER N
      F(X)=(A*X+B)*X*X+C*X+D
      A=COEFF(1)
      B=COEFF(2)
      C=COEFF(3)
      D=COEFF(4)
      R=-D/C
C
      IF(A.EQ.0D0)GO TO 1
CCC      IF(DLOG10(A)+3D0*DLOG10(R).LT.-7.8D1)GO TO 1
      IF(B.EQ.0D0 .AND. C.EQ.0D0)THEN
        R=(-D/A)**(1D0/3D0)
        RETURN
      END IF
C
   10 Z=R
      N=0
      Y1=F(R)
   11 R=R+Z
```

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

 1462           Y2=F(R)
 1463           IF(DABS(Y2).LT.DABS(Y1))GO TO 14
 1464           Y1=Y2
 1465           Z=-Z
 1466           GO TO 11
 1467        12 Y1=Y2
 1468           R=R+Z
 1469           Y2=F(R)
 1470        14 IF(Y1*Y2.LE.0D0)GO TO 16
 1471           GO TO 12
 1472        16 IF(R.EQ.0D0)GO TO 17
 1473           IF(DABS(Z/R).LT.1D-4)GO TO 18
 1474        17 Z=-Z/2D0
 1475           N=N+1
 1476           IF(N.LT.100)GO TO 12
 1477           PRINT 20,COEFF
 1478        20 FORMAT(/'UNABLE TO ACHIEVE REQUIRED ACCURACY.',1P4D12.4)
 1479        18 IF(DABS(Y1).LT.DABS(Y2))R=R-Z
 1480           RETURN
 1481  C
 1482         1 IF(B.EQ.0D0)RETURN
 1483           IF(C.EQ.0D0)THEN
 1484             R=DSQRT(-D/B)
 1485             RETURN
 1486           END IF
 1487           IF(C.LT.1D37)THEN
 1488             DISC=DSQRT(C*C-4D0*B*D)
 1489             IF(DISC-C.LT.1D-10) GO TO 10
 1490             R=(-C+DISC)/(2D0*B)
 1491             RETURN
 1492           END IF
 1493  C
 1494  C  STOP EXECUTION OF PROGRAM.
 1495  C
 1496           PRINT 40,COEFF
 1497        40 FORMAT(/'NO ROOT FOUND.',1P4D12.4)
 1498           STOP
 1499           END


ROOTS  Local Symbols

Name                         Class   Type             Size   Offset

R . . . . . . . . . . . . . param                            0006
COEFF . . . . . . . . . . . param                            000a
A . . . . . . . . . . . . . local   REAL*8              8    0128
B . . . . . . . . . . . . . local   REAL*8              8    0130
C . . . . . . . . . . . . . local   REAL*8              8    0138
D . . . . . . . . . . . . . local   REAL*8              8    0140
Y1. . . . . . . . . . . . . local   REAL*8              8    0148
```

```
ROOTS  Local Symbols

Name                          Class   Type                Size   Offset

Y2. . . . . . . . . . . . . . local   REAL*8                 8    0150
N . . . . . . . . . . . . . . local   INTEGER*4              4    0158
X . . . . . . . . . . . . . . local   REAL*8                 8    015c
Z . . . . . . . . . . . . . . local   REAL*8                 8    0164
DISC. . . . . . . . . . . . . local   REAL*8                 8    016c


Global Symbols

Name                          Class   Type                Size   Offset

ALPHAC. . . . . . . . . . . . FSUBRT  ***                  ***    098c
BOTH. . . . . . . . . . . . . common  ***                  872    0000
CALC. . . . . . . . . . . . . common  ***                15704    0000
IN. . . . . . . . . . . . . . common  ***                  314    0000
INITCM. . . . . . . . . . . . FSUBRT  ***                  ***    08c5
INPUT . . . . . . . . . . . . FSUBRT  ***                  ***    49fa
INSPEC. . . . . . . . . . . . FSUBRT  ***                  ***    2edb
LOGFN . . . . . . . . . . . . FFUNCT  REAL*8               ***    0881
NWANIN. . . . . . . . . . . . extern  ***                  ***     ***
NWMTLS. . . . . . . . . . . . extern  ***                  ***     ***
OUT . . . . . . . . . . . . . common  ***               104796    0000
OUT1. . . . . . . . . . . . . common  ***                 4592    0000
OUT2. . . . . . . . . . . . . common  ***                14656    0000
OUTPUT. . . . . . . . . . . . extern  ***                  ***     ***
PPS . . . . . . . . . . . . . extern  ***                  ***     ***
ROOTS . . . . . . . . . . . . FSUBRT  ***                  ***    59d6
SPCALC. . . . . . . . . . . . extern  ***                  ***     ***
TERM. . . . . . . . . . . . . FFUNCT  REAL*8               ***    083a
W . . . . . . . . . . . . . . common  ***                   64    0000
WFOOT . . . . . . . . . . . . extern  ***                  ***     ***
WHEAD . . . . . . . . . . . . extern  ***                  ***     ***
WPARMS. . . . . . . . . . . . FSUBRT  ***                  ***    4fa7
main. . . . . . . . . . . . . FSUBRT  ***                  ***    0000

Code size = 5d84 (23940)
Data size = 14c6 (5318)
Bss size  = 0174 (372)

No errors detected
```

```
      SUBROUTINE NWMTLS(NMVALS,NAVALS,PH,MTL,MTLSLD,KA,ITER)
C     -------------------
      IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
      INTEGER NMVALS,NAVALS,ITER,ANION,METAL,SOLID
      REAL*8 MTL(14),KA(356),COEFF(4),PH
      INTEGER MTLSLD(14)
C
      REAL*8 JC,JQ,JL,KC,KQ,KL,LC,LQ,LL
      REAL*8 SPSUM(15,14,9),HSP(15),HSPQ(15),
     &       XANIN(15),XMTL(14),XPREV(14)
      COMMON /CALC/SPSUM,HSP,HSPQ,XANIN,XMTL,XPREV
C
C
C     THIS BLOCK CALLS THE PROPER ROUTINE TO CALCULATE THE
C     NEW CONCENTRATION OF THE METALS.
C
      DO 75 METAL=1,NMVALS
        IF(MTL(METAL).EQ.0D0 .AND. MTLSLD(METAL).EQ.0)GO TO 75
        IF(MTLSLD(METAL).NE.0 .AND. ITER.EQ.1)GO TO 75
C
C  SAVE THE PREVIOUS CONCENTRATION VALUE.
C
        XPREV(METAL)=XMTL(METAL)
C
C  CALL ROUTINE TO CALCULATE NEW CONCENTRATION.
C
        SOLID=MTLSLD(METAL)
        !DO CASE SOLID
        IF (SOLID.EQ.1) THEN
! CADMIUM
          XMTL(7)=KA(40)/XANIN(1)**2
        ELSE IF (SOLID.EQ.2) THEN
! COPPER
          XMTL(8)=KA(20)/XANIN(1)**2
        ELSE IF (SOLID.EQ.3) THEN
! COPPER
          XMTL(8)=KA(30)/XANIN(1)**2
        ELSE IF (SOLID.EQ.4) THEN
! COPPER
          XMTL(8)=(KA(69)/(XANIN(2)*XANIN(1)**6))**0.25D0
        ELSE IF (SOLID.EQ.5) THEN
! MAGNESIUM
          XMTL(2)=KA(7)/(XMTL(3)*XANIN(5)**2)
        ELSE IF (SOLID.EQ.6) THEN
! CALCIUM
          XMTL(3)=KA(6)/XANIN(5)
        ELSE IF (SOLID.EQ.7) THEN
! CADMIUM
          XMTL(7)=KA(62)/XANIN(5)
        ELSE IF (SOLID.EQ.8) THEN
! COPPER
```

```
          XMTL(8)=KA(39)/XANIN(5)
        ELSE IF (SOLID.EQ.9) THEN
! COPPER
          XMTL(8)=DSQRT(KA(65)/(XANIN(5)*XANIN(1)**2))
        ELSE IF (SOLID.EQ.10) THEN
! COPPER
          XMTL(8)=(KA(66)/(XANIN(5)*XANIN(1))**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.11) THEN
! COPPER
          XMTL(8)=DSQRT(KA(70)/(XANIN(6)*XANIN(1)**3))
        ELSE IF (SOLID.EQ.12) THEN
! MAGNESIUM
          XMTL(2)=(KA(82)/(XANIN(11)**3*XANIN(1)))**0.20D0
        ELSE IF (SOLID.EQ.13) THEN
! MAGNESIUM
          XMTL(2)=KA(83)/XANIN(11)
        ELSE IF (SOLID.EQ.14) THEN
! MAGNESIUM
          XMTL(2)=(KA(84)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.15) THEN
! MAGNESIUM
          XMTL(2)=(KA(85)*10D0**PH/XANIN(11)**3)**0.25D0
        ELSE IF (SOLID.EQ.16) THEN
! CALCIUM
          XMTL(3)=(KA(75)/(XANIN(11)**3*XANIN(1)))**0.20D0
        ELSE IF (SOLID.EQ.17) THEN
! CALCIUM
          XMTL(3)=KA(76)*10D0**PH/(KA(55)*XANIN(11))
        ELSE IF (SOLID.EQ.18) THEN
! CALCIUM
          XMTL(3)=(KA(77)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.19) THEN
! CALCIUM
          XMTL(3)=(KA(78)*10D0**PH/XANIN(11)**3)**0.25D0
        ELSE IF (SOLID.EQ.20) THEN
! CADMIUM
          XMTL(7)=(KA(68)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.21) THEN
! COPPER
          XMTL(8)=(KA(67)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.22) THEN
! CADMIUM
          XMTL(7)=KA(61)/XANIN(13)
        ELSE IF (SOLID.EQ.23) THEN
! COPPER
          XMTL(8)=KA(63)/XANIN(13)
        ELSE IF (SOLID.EQ.24) THEN
! IRON(II)
          XMTL(9)=KA(94)/XANIN(13)
        ELSE IF (SOLID.EQ.25) THEN
! IRON(II)
```

```
          XMTL(9)=KA(95)/XANIN(1)**2
        ELSE IF (SOLID.EQ.26) THEN
! IRON(II)
          XMTL(9)=(KA(96)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.27) THEN
! IRON(II)
          XMTL(9)=KA(97)/XANIN(5)
        ELSE IF (SOLID.EQ.28) THEN
! MERCURY(II)
          XMTL(6)=KA(120)/XANIN(1)
        ELSE IF (SOLID.EQ.29) THEN
! MERCURY(II)
          XMTL(6)=KA(121)/XANIN(13)
        ELSE IF (SOLID.EQ.30) THEN
! MERCURY(II)
          XMTL(6)=KA(122)/XANIN(13)
        ELSE IF (SOLID.EQ.31) THEN
! MERCURY(II)
          XMTL(6)=KA(129)/XANIN(8)
        ELSE IF (SOLID.EQ.32) THEN
! COPPER
          XMTL(8)=KA(130)/XANIN(8)
        ELSE IF (SOLID.EQ.33) THEN
! MAGNESIUM
          XMTL(2)=KA(131)/XANIN(8)
        ELSE IF (SOLID.EQ.34) THEN
! MAGNESIUM
          XMTL(2)=DSQRT(KA(200)/(XANIN(9)*KA(190)*10D0**(-PH))**3)
        ELSE IF (SOLID.EQ.35) THEN
! MANGANESE(II)
          XMTL(10)=KA(201)/XANIN(1)**2
        ELSE IF (SOLID.EQ.36) THEN
! MANGANESE(II)
          XMTL(10)=KA(202)/XANIN(5)
        ELSE IF (SOLID.EQ.37) THEN
! MANGANESE(II)
          XMTL(10)=KA(203)/XANIN(13)
        ELSE IF (SOLID.EQ.38) THEN
! MANGANESE(II)
          XMTL(10)=KA(204)/XANIN(13)
        ELSE IF (SOLID.EQ.39) THEN
! MANGANESE(II)
          XMTL(10)=KA(205)/XANIN(8)
        ELSE IF (SOLID.EQ.40) THEN
! ZINC
          XMTL(11)=KA(206)/XANIN(1)**2
        ELSE IF (SOLID.EQ.41) THEN
! ZINC
          XMTL(11)=KA(207)/XANIN(1)
        ELSE IF (SOLID.EQ.42) THEN
! ZINC
```

```
          XMTL(11)=KA(208)/XANIN(5)
        ELSE IF (SOLID.EQ.43) THEN
! ZINC
          XMTL(11)=(KA(209)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.44) THEN
! ZINC
          XMTL(11)=KA(210)/XANIN(13)
        ELSE IF (SOLID.EQ.45) THEN
! CALCIUM
          XMTL(3)=KA(211)/XANIN(9)
        ELSE IF (SOLID.EQ.46) THEN
! MERCURY(I)
          XMTL(5)=KA(216)/XANIN(5)
        ELSE IF (SOLID.EQ.47) THEN
! IRON(III)
          XMTL(4)=KA(217)/XANIN(11)
        ELSE IF (SOLID.EQ.48) THEN
! MERCURY(I)
          XMTL(5)=KA(218)*10D0**PH/(KA(55)*XANIN(11))
        ELSE IF (SOLID.EQ.49) THEN
! MERCURY(I)
          XMTL(5)=KA(219)/XANIN(2)
        ELSE IF (SOLID.EQ.50) THEN
! MERCURY(I)
          XMTL(5)=KA(212)/XANIN(6)
        ELSE IF (SOLID.EQ.51) THEN
! IRON(III)
          XMTL(4)=KA(214)/XANIN(1)**3
        ELSE IF (SOLID.EQ.52) THEN
! IRON(III)
          XMTL(4)=KA(213)/XANIN(1)**3
        ELSE IF (SOLID.EQ.53) THEN
! IRON(III)
          XMTL(4)=KA(215)/XANIN(1)**3
        ELSE IF (SOLID.EQ.54) THEN
! COBALT
          XMTL(13)=KA(298)/XANIN(5)
        ELSE IF (SOLID.EQ.55) THEN
! COBALT
          XMTL(13)=KA(299)/XANIN(1)**2
        ELSE IF (SOLID.EQ.56) THEN
! COBALT
          XMTL(13)=(KA(300)/XANIN(11)**2)**(1.0/3.0)
        ELSE IF (SOLID.EQ.57) THEN
! COBALT
          XMTL(13)=KA(301)/XANIN(13)
        ELSE IF (SOLID.EQ.58) THEN
! ALUMINUM
          XMTL(14)=KA(338)/XANIN(1)**3
        ELSE IF (SOLID.EQ.59) THEN
! ALUMINUM
```

```
          XMTL(14)=KA(339)/XANIN(1)**3
        ELSE IF (SOLID.EQ.60) THEN
! ALUMINUM
          XMTL(14)=KA(340)/XANIN(11)
        ELSE IF (SOLID.EQ.61) THEN
! ALUMINUM
          XMTL(14)=KA(341)/XANIN(11)
        ELSE IF (SOLID.EQ.62) THEN
! MAGNESIUM
          XMTL(2)=KA(355)/XANIN(15)**2
        ELSE IF (SOLID.EQ.63) THEN
! CALCIUM
          XMTL(3)=KA(356)/XANIN(15)**2
        ELSE
C
C     THIS BLOCK CALCULATES THE NEW CONCENTRATION OF A METAL
C     NOT CONTROLLED BY A SOLID.
C
C  CALCULATE THE COEFFICIENTS OF THE CUBIC EQUATION.
C
          COEFF(1) = SPSUM(1,METAL,7)
          COEFF(2) = SPSUM(1,METAL,4)
          COEFF(3) = SPSUM(1,METAL,1)
          COEFF(4) = -MTL(METAL)
C         PRINT 701,ITER,METAL,COEFF
C 701     FORMAT(' SXMTL ITER=',I3,'  METAL=',I2,1P,4D12.4)
          DO 70 ANION=2,NAVALS
            IF(XANIN(ANION).GT.1D-56)THEN
              JC=SPSUM(ANION,METAL,9)
              JQ=SPSUM(ANION,METAL,8)
              JL=SPSUM(ANION,METAL,7)
              COEFF(1) = COEFF(1)+TERM(XANIN(ANION),JC,JQ,JL)
              KC=SPSUM(ANION,METAL,6)
              KQ=SPSUM(ANION,METAL,5)
              KL=SPSUM(ANION,METAL,4)
              COEFF(2) = COEFF(2)+TERM(XANIN(ANION),KC,KQ,KL)
              LC=SPSUM(ANION,METAL,3)
              LQ=SPSUM(ANION,METAL,2)
              LL=SPSUM(ANION,METAL,1)
              COEFF(3) = COEFF(3)+TERM(XANIN(ANION),LC,LQ,LL)
C             PRINT 702,ANION,COEFF,XANIN(ANION),
C    &        TERM(XANIN(ANION),JC,JQ,JL),TERM(XANIN(ANION),KC,KQ,KL),
C    &        TERM(XANIN(ANION),LC,LQ,LL)
C 702         FORMAT(' A=',I2,1P,4D12.4,4X,D12.4,4X,3D12.4)
            END IF
   70     CONTINUE
          COEFF(1) = 3D0*COEFF(1)
          COEFF(2) = 2D0*COEFF(2)
          COEFF(3) = 1D0+COEFF(3)
C
C  GET THE ROOTS OF THE CUBIC QUADRATIC OR LINEAR EQUATION.
```

```
Line# Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  256  C
  257            CALL ROOTS(COEFF,XMTL(METAL))
  258  C         PRINT 703,COEFF,XMTL(METAL)
  259  C 703     FORMAT(' ',1P,4D12.4,4X,D12.4)
  260          END IF
  261     75 CONTINUE
  262        RETURN
  263        END


NWMTLS  Local Symbols

Name                      Class   Type              Size   Offset

ITER. . . . . . . . . . . param                            0006
KA. . . . . . . . . . . . param                            000a
MTLSLD. . . . . . . . . . param                            000e
MTL . . . . . . . . . . . param                            0012
PH. . . . . . . . . . . . param                            0016
NAVALS. . . . . . . . . . param                            001a
NMVALS. . . . . . . . . . param                            001e
JC. . . . . . . . . . . . local   REAL*8               8   0000
KC. . . . . . . . . . . . local   REAL*8               8   0008
LC. . . . . . . . . . . . local   REAL*8               8   0010
JL. . . . . . . . . . . . local   REAL*8               8   0018
KL. . . . . . . . . . . . local   REAL*8               8   0020
LL. . . . . . . . . . . . local   REAL*8               8   0028
JQ. . . . . . . . . . . . local   REAL*8               8   0030
KQ. . . . . . . . . . . . local   REAL*8               8   0038
LQ. . . . . . . . . . . . local   REAL*8               8   0040
COEFF . . . . . . . . . . local   REAL*8              32   0048
METAL . . . . . . . . . . local   INTEGER*4            4   0068
ANION . . . . . . . . . . local   INTEGER*4            4   006c
SOLID . . . . . . . . . . local   INTEGER*4            4   0070
SPSUM . . . . . . . . . . CALC    REAL*8           15120   0000
HSP . . . . . . . . . . . CALC    REAL*8             120   3b10
HSPQ. . . . . . . . . . . CALC    REAL*8             120   3b88
XANIN . . . . . . . . . . CALC    REAL*8             120   3c00
XMTL. . . . . . . . . . . CALC    REAL*8             112   3c78
XPREV . . . . . . . . . . CALC    REAL*8             112   3ce8

  264  C
  265        SUBROUTINE NWANIN(NMVALS,NAVALS,PH,A,ASLD,KA,
  266       &                               CONVTM,S$PCO2,BETA)
  267  C     ------------------
  268        IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
  269        REAL*8 A(15),KA(356),COEFF(4),CONVTM(14),PH
  270        LOGICAL*2 S$PCO2
  271        INTEGER NAVALS,NMVALS,SOLID,ANION
  272        INTEGER ASLD(15)
  273  C
```

```
      REAL*8 JC,JQ,JL,KC,KQ,KL,LC,LQ,LL,CONV,BETA
      REAL*8 SPSUM(15,14,9),HSP(15),HSPQ(15),
     &       XANIN(15),XMTL(14),XPREV(14)
      COMMON /CALC/SPSUM,HSP,HSPQ,XANIN,XMTL,XPREV
C
C
C       THIS BLOCK CALLS THE PROPER ROUTINE TO CALCULATE THE
C       NEW CONCENTRATION OF THE ANIONS.
C
      DO 95 ANION=2,NAVALS
        IF(A(ANION).EQ.0D0 .AND. ASLD(ANION).EQ.0)GO TO 95
        IF(ANION.EQ.5 .AND. S$PCO2) GO TO 95
C
C  CALL ROUTINE TO CALCULATE NEW CONCENTRATION.
C
        SOLID=ASLD(ANION)
        !DO CASE SOLID
        IF (SOLID.EQ.1) THEN
        !ELSE IF (SOLID.EQ.2) THEN
        !ELSE IF (SOLID.EQ.3) THEN
        ELSE IF (SOLID.EQ.4) THEN
 ! SULFATE
          XANIN(2)=KA(69)/(XMTL(8)**4*XANIN(1)**6)
        ELSE IF (SOLID.EQ.5) THEN
 ! INORGANIC CARBON
          XANIN(5)=DSQRT(KA(7)/(XMTL(2)*XMTL(3)))
        ELSE IF (SOLID.EQ.6) THEN
 ! INORGANIC CARBON
          XANIN(5)=KA(6)/XMTL(3)
        ELSE IF (SOLID.EQ.7) THEN
 ! INORGANIC CARBON
          XANIN(5)=KA(62)/XMTL(7)
        ELSE IF (SOLID.EQ.8) THEN
 ! INORGANIC CARBON
          XANIN(5)=KA(39)/XMTL(8)
        ELSE IF (SOLID.EQ.9) THEN
 ! INORGANIC CARBON
          XANIN(5)=KA(65)/(XMTL(8)**2*XANIN(1)**2)
        ELSE IF (SOLID.EQ.10) THEN
 ! INORGANIC CARBON
          XANIN(5)=DSQRT(KA(66)/(XMTL(8)**3*XANIN(1)**2))
        ELSE IF (SOLID.EQ.11) THEN
 ! CHLORIDE
          XANIN(6)=KA(70)/(XMTL(8)**2*XANIN(1)**3)
        ELSE IF (SOLID.EQ.12) THEN
 ! ORTHO-PHOSPHATE
          XANIN(11)=(KA(82)/(XMTL(2)**5*XANIN(1)))**(1.0/3.0)
        ELSE IF (SOLID.EQ.13) THEN
 ! ORTHO-PHOSPHATE
          XANIN(11)=KA(83)/XMTL(2)
        ELSE IF (SOLID.EQ.14) THEN
```

```
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(84)/XMTL(2)**3)
        ELSE IF (SOLID.EQ.15) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=(KA(85)*10D0**PH/XMTL(2)**4)**(1.0/3.0)
        ELSE IF (SOLID.EQ.16) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=(KA(75)/(XMTL(3)**5*XANIN(1)))**(1.0/3.0)
        ELSE IF (SOLID.EQ.17) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=KA(76)*10D0**PH/(KA(55)*XMTL(3))
        ELSE IF (SOLID.EQ.18) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(77)/XMTL(3)**3)
        ELSE IF (SOLID.EQ.19) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=(KA(78)*10D0**PH/XMTL(3)**4)**(1.0/3.0)
        ELSE IF (SOLID.EQ.20) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(68)/XMTL(7)**3)
        ELSE IF (SOLID.EQ.21) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(67)/XMTL(8)**3)
        ELSE IF (SOLID.EQ.22) THEN
! SULFIDE
          XANIN(13)=KA(61)/XMTL(7)
        ELSE IF (SOLID.EQ.23) THEN
! SULFIDE
          XANIN(13)=KA(63)/XMTL(8)
        ELSE IF (SOLID.EQ.24) THEN
! SULFIDE
          XANIN(13)=KA(94)/XMTL(9)
        !ELSE IF (SOLID.EQ.25) THEN
        ELSE IF (SOLID.EQ.26) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(96)/XMTL(9)**3)
        ELSE IF (SOLID.EQ.27) THEN
! INORGANIC CARBON
          XANIN(5)=KA(97)/XMTL(9)
        !ELSE IF (SOLID.EQ.28) THEN
        ELSE IF (SOLID.EQ.29) THEN
! SULFIDE
          XANIN(13)=KA(121)/XMTL(6)
        ELSE IF (SOLID.EQ.30) THEN
! SULFIDE
          XANIN(13)=KA(122)/XMTL(6)
        ELSE IF (SOLID.EQ.31) THEN
! SELENITE
          XANIN(8)=KA(129)/XMTL(6)
        ELSE IF (SOLID.EQ.32) THEN
! SELENITE
```

```
          XANIN(8)=KA(130)/XMTL(8)
        ELSE IF (SOLID.EQ.33) THEN
! SELENITE
          XANIN(8)=KA(131)/XMTL(2)
        ELSE IF (SOLID.EQ.34) THEN
! SILICATE
          XANIN(9)=(KA(200)/(XMTL(2)**2*(KA(190)*10D0**(-PH))**3))
     &                                                      **(1.0/3.0)
        !ELSE IF (SOLID.EQ.35) THEN
        ELSE IF (SOLID.EQ.36) THEN
! INORGANIC CARBON
          XANIN(5)=KA(202)/XMTL(10)
        ELSE IF (SOLID.EQ.37) THEN
! SULFIDE
          XANIN(13)=KA(203)/XMTL(10)
        ELSE IF (SOLID.EQ.38) THEN
! SULFIDE
          XANIN(13)=KA(204)/XMTL(10)
        ELSE IF (SOLID.EQ.39) THEN
! SELENITE
          XANIN(8)=KA(205)/XMTL(10)
        !ELSE IF (SOLID.EQ.40) THEN
        !ELSE IF (SOLID.EQ.41) THEN
        ELSE IF (SOLID.EQ.42) THEN
! INORGANIC CARBON
          XANIN(5)=KA(208)/XMTL(11)
        ELSE IF (SOLID.EQ.43) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(209)/XMTL(11)**3)
        ELSE IF (SOLID.EQ.44) THEN
! SULFIDE
          XANIN(13)=KA(210)/XMTL(11)
        ELSE IF (SOLID.EQ.45) THEN
! SILICATE
          XANIN(9)=KA(211)/XMTL(3)
        ELSE IF (SOLID.EQ.46) THEN
! INORGANIC CARBON
          XANIN(5)=KA(216)/XMTL(5)
        ELSE IF (SOLID.EQ.47) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=KA(217)/XMTL(4)
        ELSE IF (SOLID.EQ.48) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=KA(218)*10D0**PH/(KA(55)*XMTL(5))
        ELSE IF (SOLID.EQ.49) THEN
! SULFATE
          XANIN(2)=KA(219)/XMTL(5)
        ELSE IF (SOLID.EQ.50) THEN
! CHLORIDE
          XANIN(6)=KA(212)/XMTL(5)
        !ELSE IF (SOLID.EQ.51) THEN
        !ELSE IF (SOLID.EQ.52) THEN
        !ELSE IF (SOLID.EQ.53) THEN
        ELSE IF (SOLID.EQ.54) THEN
! INORGANIC CARBON
          XANIN(5)=KA(298)/XMTL(13)
        !ELSE IF (SOLID.EQ.55) THEN
        ELSE IF (SOLID.EQ.56) THEN
! ORTHO-PHOSPHATE
          XANIN(11)=DSQRT(KA(300)/XMTL(13)**3)
        ELSE IF (SOLID.EQ.57) THEN
! SULFIDE
          XANIN(13)=KA(301)/XMTL(13)
        !ELSE IF (SOLID.EQ.58) THEN
        !ELSE IF (SOLID.EQ.59) THEN
        ELSE IF (SOLID.EQ.60) THEN
! PHOSPHATE
          XANIN(11)=KA(340)/XMTL(14)
        ELSE IF (SOLID.EQ.61) THEN
! PHOSPHATE
          XANIN(11)=KA(341)/XMTL(14)
        ELSE IF (SOLID.EQ.62) THEN
! FLUORIDE
          XANIN(15)=KA(355)/XMTL(2)
        ELSE IF (SOLID.EQ.63) THEN
! FLUORIDE
          XANIN(15)=KA(356)/XMTL(3)
        ELSE
C
C     THIS BLOCK CALCULATES THE NEW CONCENTRATION OF AN ANION
C     NOT CONTROLLED BY A SOLID.
C
C  CALCULATE THE COEFFICIENTS OF THE CUBIC EQUATION.
C
          COEFF(1) = 0D0
          COEFF(2) = HSPQ(ANION)
          COEFF(3) = HSP(ANION)
          COEFF(4) = -A(ANION)
C         PRINT 704,ITER,ANION,COEFF
C 704     FORMAT(' SXANIN ITER=',I3,'  ANION=',I2,1P,4D12.4)
          DO 90 METAL=1,NMVALS
C
C FOR SS CALCULATION, CONVERT METALS TO MG/L
C     BETA IS (TOTAL SEDIMENT) / (FREE SEDIMENT)
C
            IF (ANION .EQ. 14) THEN
              CONV=CONVTM(METAL)*BETA
            ELSE
              CONV=1D0
            END IF
            IF(XMTL(METAL).GT.1D-56)THEN
              JC=SPSUM(ANION,METAL,9)
```

```
Line#  Source Line        Microsoft FORTRAN Optimizing Compiler Version 5.00

  478                   JQ=SPSUM(ANION,METAL,6)
  479                   JL=SPSUM(ANION,METAL,3)
  480                   COEFF(1) = COEFF(1)+TERM(XMTL(METAL)*CONV,JC,JQ,JL)
  481                   KC=SPSUM(ANION,METAL,8)
  482                   KQ=SPSUM(ANION,METAL,5)
  483                   KL=SPSUM(ANION,METAL,2)
  484                   COEFF(2) = COEFF(2)+TERM(XMTL(METAL)*CONV,KC,KQ,KL)
  485                   LC=SPSUM(ANION,METAL,7)
  486                   LQ=SPSUM(ANION,METAL,4)
  487                   LL=SPSUM(ANION,METAL,1)
  488                   COEFF(3) = COEFF(3)+TERM(XMTL(METAL)*CONV,LC,LQ,LL)
  489   C               PRINT 705,METAL,COEFF,XMTL(METAL),
  490   C    &          TERM(XMTL(METAL),JC,JQ,JL),TERM(XMTL(METAL),KC,KQ,KL),
  491   C    &          TERM(XMTL(METAL),LC,LQ,LL)
  492   C 705           FORMAT(' M=',I2,1P,4D12.4,4X,D12.4,4X,3D12.4)
  493                 END IF
  494      90       CONTINUE
  495               COEFF(1) = 3D0*COEFF(1)
  496               COEFF(2) = 2D0*COEFF(2)
  497               COEFF(3) = 1D0+COEFF(3)
  498   C
  499   C  GET THE ROOTS OF THE CUBIC QUADRATIC OR LINEAR EQUATION.
  500   C
  501               CALL ROOTS(COEFF,XANIN(ANION))
  502   C
  503   C           PRINT 706,COEFF,XANIN(ANION)
  504   C 706       FORMAT(' ',1P,4D12.4,4X,D12.4)
  505   C
  506             END IF
  507      95 CONTINUE
  508   C      IF (XANIN(14).NE.0D0) THEN
  509   C        BETA=A(14)/XANIN(14)
  510   C      END IF
  511         RETURN
  512         END


NWANIN  Local Symbols

Name                          Class   Type              Size   Offset

BETA. . . . . . . . . . . . . param                              0006
S$PC02. . . . . . . . . . . . param                              000a
CONVTM. . . . . . . . . . . . param                              000e
KA. . . . . . . . . . . . . . param                              0012
ASLD. . . . . . . . . . . . . param                              0016
A . . . . . . . . . . . . . . param                              001a
PH. . . . . . . . . . . . . . param                              001e
NAVALS. . . . . . . . . . . . param                              0022
NMVALS. . . . . . . . . . . . param                              0026
JC. . . . . . . . . . . . . . local   REAL*8               8     0074
```

```
NWANIN  Local Symbols

Name                          Class   Type              Size   Offset

KC. . . . . . . . . . . . . . local   REAL*8               8    007c
LC. . . . . . . . . . . . . . local   REAL*8               8    0084
JL. . . . . . . . . . . . . . local   REAL*8               8    008c
KL. . . . . . . . . . . . . . local   REAL*8               8    0094
LL. . . . . . . . . . . . . . local   REAL*8               8    009c
JQ. . . . . . . . . . . . . . local   REAL*8               8    00a4
KQ. . . . . . . . . . . . . . local   REAL*8               8    00ac
LQ. . . . . . . . . . . . . . local   REAL*8               8    00b4
COEFF . . . . . . . . . . . . local   REAL*8              32    00bc
METAL . . . . . . . . . . . . local   REAL*8               8    00dc
ANION . . . . . . . . . . . . local   INTEGER*4            4    00e4
CONV. . . . . . . . . . . . . local   REAL*8               8    00e8
SOLID . . . . . . . . . . . . local   INTEGER*4            4    00f0
SPSUM . . . . . . . . . . . . CALC    REAL*8           15120    0000
HSP . . . . . . . . . . . . . CALC    REAL*8             120    3b10
HSPQ. . . . . . . . . . . . . CALC    REAL*8             120    3b88
XANIN . . . . . . . . . . . . CALC    REAL*8             120    3c00
XMTL. . . . . . . . . . . . . CALC    REAL*8             112    3c78
XPREV . . . . . . . . . . . . CALC    REAL*8             112    3ce8
```

```
C
      SUBROUTINE SPCALC(PH,NUMPH,NMVALS,NAVALS,BETA,SSCDB)
C     -------------------
      IMPLICIT REAL*8(A-H,K-Z),INTEGER(I-J)
      REAL*8 PH,H,H2,H3,H4,X1,X2,X3,BETA,SSCDB
      INTEGER NUMPH,NMVALS,NAVALS,I
      REAL*8 X(319,41)
      REAL*4 PHS(41),TDSO4(41),TDEDTA(41),TDNTA(41),
     &TDIC(41),TDCL(41),TDOC(41),TDSE(41),TDSIO4(41),TDNH3(41),
     &TDPO4(41),TDAC(41),TDS(41),TDNA(41),TDMG(41),TDCA(41),TDFE3(41),
     &TDHG1(41),TDHG2(41),TDCD(41),TDCU(41),TDFE2(41),TDMN2(41),
     &TDZN(41),TDK(41),TDCO(41),TDSS(41),TDAL(41),TDF(41)
      COMMON /OUT/X,PHS
      COMMON /OUT1/TDSO4,TDEDTA,TDNTA,TDIC,TDCL,TDOC,
     &             TDSE,TDSIO4,TDNH3,TDPO4,TDAC,TDS,TDSS,TDF,
     &             TDNA,TDMG,TDCA,TDFE3,TDHG1,TDHG2,TDCD,TDCU,
     &             TDFE2,TDMN2,TDZN,TDK,TDCO,TDAL
      REAL*8 C(356),M(356),ALPHA(356),KA(356),GAM(319)
      CHARACTER KTAG(356)*2
      COMMON /OUT2/C,M,ALPHA,KA,GAM,KTAG
C
      REAL*8 SPSUM(15,14,9),HSP(15),HSPQ(15),
     &       XANIN(15),XMTL(14),XPREV(14)
      COMMON /CALC/SPSUM,HSP,HSPQ,XANIN,XMTL,XPREV
      REAL*8 XA(3,15)/45*0D0/,XM(3,14)/42*0D0/
C
```

```
   539  C
#  540  C     THIS BLOCK CALCULATES AND STORES THE CONCENTRATIONS OF ALL CHEMIC
#  541  C     SPECIES IN THE SYSTEM AT THE CURRENT PH.  AFTER THE SYSTEM HAS BE
   542  C     EVALUATED AT EVERY PH VALUE THE RESULTS ARE PRINTED OUT.
   543  C
   544        H  = 10D0**(-PH)
***** NWS.FOR(544) : warning F4063: SPCALC : function too large for post-optimi
zer
   545        H2 = H*H
   546        H3 = H*H2
   547        H4 = H*H3
   548  C
   549        X(  1,NUMPH) = XANIN(5)
   550        X(  2,NUMPH) = XANIN(5)/KA(2)*H
   551        X(  3,NUMPH) = XMTL(3)
   552        X(  4,NUMPH) = XANIN(5)/(KA(1)*KA(2))*H2
   553        X(  5,NUMPH) = XMTL(2)
   554        X(  6,NUMPH) = XANIN(2)
   555        X(  7,NUMPH) = XANIN(5)/KA(4)*XMTL(3)
   556        X(  8,NUMPH) = XANIN(2)/KA(3)*XMTL(3)
   557        X(  9,NUMPH) = XANIN(5)/KA(5)*XMTL(2)
   558        X( 10,NUMPH) = XANIN(2)/KA(10)*XMTL(2)
   559        X( 11,NUMPH) = XANIN(5)/(KA(2)*KA(8))*H*XMTL(3)
   560        X( 12,NUMPH) = XANIN(5)/(KA(2)*KA(9))*H*XMTL(2)
   561        X( 13,NUMPH) = XANIN(1)/KA(13)*XMTL(3)
   562        X( 14,NUMPH) = XANIN(1)/KA(14)*XMTL(2)
   563        X( 15,NUMPH) = XANIN(2)*KA(15)*H
   564        X( 16,NUMPH) = XANIN(6)
   565        X( 17,NUMPH) = XANIN(1)
   566        X( 18,NUMPH) = XMTL(8)
   567        X( 19,NUMPH) = XANIN(5)*KA(18)*XMTL(8)
   568        X( 20,NUMPH) = XANIN(5)**2*KA(19)*XMTL(8)
   569        X( 21,NUMPH) = KA(21)/H*XMTL(8)
   570        X( 22,NUMPH) = KA(22)/H2*XMTL(8)
   571        X( 23,NUMPH) = KA(23)/H3*XMTL(8)
   572        X( 24,NUMPH) = KA(24)/H4*XMTL(8)
   573        X( 25,NUMPH) = KA(25)/H2*XMTL(8)**2
   574        X( 26,NUMPH) = XANIN(2)*KA(26)*XMTL(8)
   575        X( 27,NUMPH) = XANIN(5)*KA(27)*H*XMTL(8)
   576        X( 28,NUMPH) = XANIN(6)*KA(28)*XMTL(8)
   577        X( 29,NUMPH) = XANIN(6)**2*KA(29)*XMTL(8)
   578        X( 30,NUMPH) = XANIN(7)**2*KA(34)/H2*XMTL(8)
   579        X( 31,NUMPH) = XANIN(10)*KA(31)*XMTL(8)
   580        X( 32,NUMPH) = XANIN(10)**2*KA(32)*XMTL(8)
   581        X( 33,NUMPH) = XANIN(10)**3*KA(33)*XMTL(8)
   582        X( 34,NUMPH) = XANIN(10)*H/KA(35)
   583        X( 35,NUMPH) = XANIN(10)
   584        X( 36,NUMPH) = XANIN(12)*KA(36)*XMTL(8)
   585        X( 37,NUMPH) = XANIN(12)**2*KA(37)*XMTL(8)
   586        X( 38,NUMPH) = XANIN(12)
```

```
      X( 39,NUMPH) = XANIN(12)*KA(38)*H
      X( 40,NUMPH) = XMTL(7)
      X( 41,NUMPH) = KA(41)/H*XMTL(7)
      X( 42,NUMPH) = KA(42)/H2*XMTL(7)
      X( 43,NUMPH) = KA(43)/H3*XMTL(7)
      X( 44,NUMPH) = KA(44)/H4*XMTL(7)
      X( 45,NUMPH) = KA(45)/H*XMTL(7)**2
      X( 46,NUMPH) = XANIN(2)*KA(46)*XMTL(7)
      X( 47,NUMPH) = XANIN(6)*KA(47)*XMTL(7)
      X( 48,NUMPH) = XANIN(6)**2*KA(48)*XMTL(7)
      X( 49,NUMPH) = XANIN(6)**3*KA(49)*XMTL(7)
      X( 50,NUMPH) = XANIN(10)*KA(50)*XMTL(7)
      X( 51,NUMPH) = XANIN(10)**2*KA(51)*XMTL(7)
      X( 52,NUMPH) = XANIN(10)**3*KA(52)*XMTL(7)
      X( 53,NUMPH) = XANIN(11)*KA(55)*KA(54)*KA(53)*H3
      X( 54,NUMPH) = XANIN(11)*KA(55)*KA(54)*H2
      X( 55,NUMPH) = XANIN(11)*KA(55)*H
      X( 56,NUMPH) = XANIN(12)*KA(56)*XMTL(7)
      X( 57,NUMPH) = XANIN(12)**2*KA(57)*XMTL(7)
      X( 58,NUMPH) = XANIN(12)**3*KA(58)*XMTL(7)
      X( 59,NUMPH) = XANIN(13)*KA(60)*KA(59)*H2
      X( 60,NUMPH) = XANIN(13)*KA(60)*H
      X( 61,NUMPH) = XANIN(13)
      X( 62,NUMPH) = XANIN(11)
      X( 63,NUMPH) = XANIN(7)**2*KA(71)/H2*XMTL(7)
      X( 64,NUMPH) = XANIN(11)*KA(72)*XMTL(3)
      X( 65,NUMPH) = XANIN(11)*KA(55)*H*KA(73)*XMTL(3)
      X( 66,NUMPH) = XANIN(11)*KA(55)*KA(54)*H2*KA(74)*XMTL(3)
      X( 67,NUMPH) = XANIN(11)*KA(79)*XMTL(2)
      X( 68,NUMPH) = XANIN(11)*KA(55)*H*KA(80)*XMTL(2)
      X( 69,NUMPH) = XANIN(11)*KA(55)*KA(54)*H2*KA(81)*XMTL(2)
      X( 70,NUMPH) = XANIN(2)*H2*KA(15)*KA(16)
      X( 71,NUMPH) = XANIN(7)
      X( 72,NUMPH) = XMTL(9)
      X( 73,NUMPH) = KA(86)/H*XMTL(9)
      X( 74,NUMPH) = KA(87)/H2*XMTL(9)
      X( 75,NUMPH) = KA(88)/H3*XMTL(9)
      X( 76,NUMPH) = KA(89)/H2**2*XMTL(9)
      X( 77,NUMPH) = XANIN(11)*KA(55)*H*KA(91)*XMTL(9)
      X( 78,NUMPH) = XANIN(11)*KA(55)*KA(54)*H2*KA(92)*XMTL(9)
      X( 79,NUMPH) = XANIN(12)*KA(93)*XMTL(9)
      X( 80,NUMPH) = XANIN(2)*KA(90)*XMTL(9)
      X( 81,NUMPH) = XANIN(13)*KA(60)*H*KA(98)*XMTL(7)
      X( 82,NUMPH) = (XANIN(13)*KA(60)*H)**2*KA(99)*XMTL(7)
      X( 83,NUMPH) = (XANIN(13)*KA(60)*H)**3*KA(100)*XMTL(7)
      X( 84,NUMPH) = XANIN(1)*KA(101)*XMTL(6)
      X( 85,NUMPH) = XANIN(1)**2*KA(102)*XMTL(6)
      X( 86,NUMPH) = XANIN(1)**3*KA(103)*XMTL(6)
      X( 87,NUMPH) = XANIN(1)*KA(104)*XMTL(6)*XMTL(6)
      X( 88,NUMPH) = XANIN(1)**3*KA(105)*XMTL(6)*XMTL(6)**2
      X( 89,NUMPH) = XANIN(6)*XANIN(1)*KA(106)*XMTL(6)
```

```
 638        X( 90,NUMPH) = XANIN(6)*KA(107)*XMTL(6)
 639        X( 91,NUMPH) = XANIN(6)**2*KA(108)*XMTL(6)
 640        X( 92,NUMPH) = XANIN(6)**3*KA(109)*XMTL(6)
 641        X( 93,NUMPH) = XANIN(10)*KA(110)*XMTL(6)
 642        X( 94,NUMPH) = XANIN(10)**2*KA(111)*XMTL(6)
 643        X( 95,NUMPH) = XANIN(10)**3*KA(112)*XMTL(6)
 644        X( 96,NUMPH) = XANIN(12)*KA(113)*XMTL(6)
 645        X( 97,NUMPH) = XANIN(12)**2*KA(114)*XMTL(6)
 646        X( 98,NUMPH) = XANIN(12)**3*KA(115)*XMTL(6)
 647        X( 99,NUMPH) = (XANIN(13)*KA(60)*H)**2*KA(116)*XMTL(6)
 648        X(100,NUMPH) = XANIN(2)*KA(117)*XMTL(6)
 649        X(101,NUMPH) = XANIN(2)**2*KA(118)*XMTL(6)
 650        X(102,NUMPH) = XANIN(7)**2*KA(119)*XMTL(6)
 651        X(103,NUMPH) = XMTL(6)
 652        X(104,NUMPH) = XANIN(8)
 653        X(105,NUMPH) = XANIN(8)*KA(123)*H
 654        X(106,NUMPH) = XANIN(8)*KA(124)*KA(123)*H2
 655        X(107,NUMPH) = (XANIN(8)*KA(123))**2*KA(125)*H2
 656        X(108,NUMPH) = (XANIN(8)*KA(123))**2*KA(125)*H/KA(126)
 657        X(109,NUMPH) = XANIN(8)**2*KA(127)*XMTL(6)
 658        X(110,NUMPH) = XANIN(8)**2*KA(128)*XMTL(7)
 659        X(111,NUMPH) = XANIN(1)*KA(134)*XMTL(10)**2
 660        X(112,NUMPH) = XANIN(1)**3*KA(135)*XMTL(10)**2
 661        X(113,NUMPH) = XANIN(5)*KA(136)*H/KA(2)*XMTL(10)
 662        X(114,NUMPH) = XANIN(2)*KA(137)*XMTL(10)
 663        X(115,NUMPH) = XANIN(1)*KA(132)*XMTL(10)
 664        X(116,NUMPH) = XANIN(1)**4*KA(133)*XMTL(10)
 665        X(117,NUMPH) = XANIN(6)*KA(138)*XMTL(10)
 666        X(118,NUMPH) = XANIN(12)*KA(139)*XMTL(10)
 667        X(119,NUMPH) = XANIN(10)*KA(140)*XMTL(10)
 668        X(120,NUMPH) = XANIN(10)**2*KA(141)*XMTL(10)
 669        X(121,NUMPH) = XANIN(10)**3*KA(142)*XMTL(10)
 670        X(122,NUMPH) = XANIN(1)*KA(143)*XMTL(11)
 671        X(123,NUMPH) = XANIN(1)**2*KA(144)*XMTL(11)
 672        X(124,NUMPH) = XANIN(1)**3*KA(145)*XMTL(11)
 673        X(125,NUMPH) = XANIN(1)*KA(146)*XMTL(11)**2
 674        X(126,NUMPH) = XANIN(11)*KA(55)*H*KA(147)*XMTL(11)
 675        X(127,NUMPH) = XANIN(11)*KA(55)*KA(54)*H2*KA(148)*XMTL(11)
 676        X(128,NUMPH) = XANIN(2)*KA(149)*XMTL(11)
 677        X(129,NUMPH) = XANIN(2)**2*KA(150)*XMTL(11)
 678        X(130,NUMPH) = XANIN(2)**3*KA(151)*XMTL(11)
 679        X(131,NUMPH) = XANIN(6)*KA(152)*XMTL(11)
 680        X(132,NUMPH) = XANIN(6)**2*KA(153)*XMTL(11)
 681        X(133,NUMPH) = XANIN(6)**3*KA(154)*XMTL(11)
 682        X(134,NUMPH) = XANIN(12)*KA(155)*XMTL(11)
 683        X(135,NUMPH) = XANIN(12)**2*KA(156)*XMTL(11)
 684        X(136,NUMPH) = XANIN(12)**3*KA(157)*XMTL(11)
 685        X(137,NUMPH) = XANIN(10)*KA(158)*XMTL(11)
 686        X(138,NUMPH) = XANIN(10)**2*KA(159)*XMTL(11)
 687        X(139,NUMPH) = XANIN(10)**3*KA(160)*XMTL(11)
 688        X(140,NUMPH) = XANIN(13)*KA(60)*H*XANIN(1)*KA(161)*XMTL(11)
```

```
      X(141,NUMPH) = XANIN(1)*KA(162)*XMTL(1)
      X(142,NUMPH) = XANIN(11)*KA(55)*H*KA(163)*XMTL(1)
      X(143,NUMPH) = XANIN(2)*KA(164)*XMTL(1)
      X(144,NUMPH) = XANIN(12)*KA(165)*XMTL(1)
      X(145,NUMPH) = XANIN(1)*KA(166)*XMTL(12)
      X(146,NUMPH) = XANIN(11)*KA(55)*H*KA(167)*XMTL(12)
      X(147,NUMPH) = XANIN(2)*KA(168)*XMTL(12)
      X(148,NUMPH) = XANIN(6)*KA(169)*XMTL(12)
      X(149,NUMPH) = XANIN(1)*KA(170)*XMTL(5)
      X(150,NUMPH) = XANIN(2)*KA(171)*XMTL(5)
      X(151,NUMPH) = XANIN(2)**2*KA(172)*XMTL(5)
      X(152,NUMPH) = XANIN(1)*KA(173)*XMTL(4)
      X(153,NUMPH) = XANIN(1)**2*KA(174)*XMTL(4)
      X(154,NUMPH) = XANIN(1)**2*KA(175)*XMTL(4)**2
      X(155,NUMPH) = XANIN(1)**4*KA(176)*XMTL(4)**3
      X(156,NUMPH) = XANIN(11)*KA(55)*H*KA(177)*XMTL(4)
      X(157,NUMPH) = XANIN(11)*KA(55)*KA(54)*H2*KA(178)*XMTL(4)
      X(158,NUMPH) = XANIN(2)*KA(179)*XMTL(4)
      X(159,NUMPH) = XANIN(2)**2*KA(180)*XMTL(4)
      X(160,NUMPH) = XANIN(6)*KA(181)*XMTL(4)
      X(161,NUMPH) = XANIN(6)**2*KA(182)*XMTL(4)
      X(162,NUMPH) = XANIN(6)**3*KA(183)*XMTL(4)
      X(163,NUMPH) = XANIN(12)*KA(184)*XMTL(4)
      X(164,NUMPH) = XANIN(12)**2*KA(185)*XMTL(4)
      X(165,NUMPH) = XANIN(12)**3*KA(186)*XMTL(4)
      X(166,NUMPH) = XANIN(8)*KA(123)*H*KA(187)*XMTL(4)
      X(167,NUMPH) = XANIN(9)*KA(190)*H*KA(188)*XMTL(4)
      X(168,NUMPH) = XANIN(9)*KA(190)*KA(189)*H2
      X(169,NUMPH) = XANIN(9)*KA(190)*H
      X(170,NUMPH) = XANIN(9)**2*KA(191)*H2
C     X(171,NUMPH) =
C     X(172,NUMPH) =
      X(173,NUMPH) = XANIN(9)*KA(194)*XMTL(2)
      X(174,NUMPH) = XANIN(9)*KA(190)*H*KA(195)*XMTL(2)
      X(175,NUMPH) = (XANIN(9)*KA(190)*H)**2*KA(196)*XMTL(2)
      X(176,NUMPH) = XANIN(9)*KA(197)*XMTL(3)
      X(177,NUMPH) = XANIN(9)*KA(190)*H*KA(198)*XMTL(3)
      X(178,NUMPH) = (XANIN(9)*KA(190)*H)**2*KA(199)*XMTL(3)
      X(179,NUMPH) = XMTL(10)
      X(180,NUMPH) = XMTL(11)
      X(181,NUMPH) = XMTL(1)
      X(182,NUMPH) = XMTL(12)
      X(183,NUMPH) = XMTL(5)
      X(184,NUMPH) = XMTL(4)
      X(185,NUMPH) = H
      X(186,NUMPH) = XANIN(9)
C     X(187,NUMPH) =
C     X(188,NUMPH) = (XANIN(13)*KA(60)*H)**2*KA(220)*XMTL(11)
      X(189,NUMPH) = XANIN(3)
      X(190,NUMPH) = XANIN(3)*KA(221)*H
      X(191,NUMPH) = XANIN(3)*KA(221)*KA(222)*H2
      X(192,NUMPH) = XANIN(3)*KA(221)*KA(222)*KA(223)*H3
      X(193,NUMPH) = XANIN(3)*KA(221)*KA(222)*KA(223)*KA(224)*H4
      X(194,NUMPH) = XANIN(3)*KA(225)*XMTL(1)
      X(195,NUMPH) = XANIN(3)*KA(226)*XMTL(2)
      X(196,NUMPH) = XANIN(3)*KA(226)*KA(227)*H*XMTL(2)
      X(197,NUMPH) = XANIN(3)*KA(228)*XMTL(3)
      X(198,NUMPH) = XANIN(3)*KA(228)*KA(229)*H*XMTL(3)
      X(199,NUMPH) = XANIN(3)*KA(230)*XMTL(4)
      X(200,NUMPH) = XANIN(3)*KA(230)*KA(231)*H*XMTL(4)
      X(201,NUMPH) = XANIN(3)*KA(230)/(KA(232)*H)*XMTL(4)
      X(202,NUMPH) = XANIN(3)*KA(230)/(KA(232)*KA(233)*H2)*XMTL(4)
      X(203,NUMPH) = (XANIN(3)*KA(230)*XMTL(4))**2/(KA(234)*H2)
      X(204,NUMPH) = XANIN(3)*KA(235)*XMTL(6)
      X(205,NUMPH) = XANIN(3)*KA(235)*KA(236)*H*XMTL(6)
      X(206,NUMPH) = XANIN(3)*KA(235)/(KA(237)*H)*XMTL(6)
      X(207,NUMPH) = XANIN(3)*KA(238)*XMTL(7)
      X(208,NUMPH) = XANIN(3)*KA(238)*KA(239)*H*XMTL(7)
      X(209,NUMPH) = XANIN(3)*KA(240)*XMTL(8)
      X(210,NUMPH) = XANIN(3)*KA(240)*KA(241)*H*XMTL(8)
      X(211,NUMPH) = XANIN(3)*XANIN(1)*KA(240)*KA(242)*XMTL(8)
      X(212,NUMPH) = XANIN(3)*KA(243)*XMTL(9)
      X(213,NUMPH) = XANIN(3)*KA(243)*KA(244)*H*XMTL(9)
      X(214,NUMPH) = XANIN(3)*KA(243)/(KA(245)*H)*XMTL(9)
      X(215,NUMPH) = XANIN(3)*KA(243)/(KA(245)*KA(246)*H2)*XMTL(9)
      X(216,NUMPH) = XANIN(3)*KA(247)*XMTL(10)
      X(217,NUMPH) = XANIN(3)*KA(247)*KA(248)*H*XMTL(10)
      X(218,NUMPH) = XANIN(3)*KA(249)*XMTL(11)
      X(219,NUMPH) = XANIN(3)*KA(249)*KA(250)*H*XMTL(11)
      X(220,NUMPH) = XANIN(3)*XANIN(1)*KA(249)*KA(251)*XMTL(11)
      X(221,NUMPH) = XANIN(3)*KA(252)*XMTL(12)
      X(222,NUMPH) = XMTL(13)
      X(223,NUMPH) = XANIN(3)*KA(253)*KA(254)*H*XMTL(13)
      X(224,NUMPH) = XANIN(3)*KA(253)*XMTL(13)
      X(225,NUMPH) = XANIN(1)*KA(255)*XMTL(13)
      X(226,NUMPH) = XANIN(1)**2*KA(256)*XMTL(13)
      X(227,NUMPH) = XANIN(1)**3*KA(257)*XMTL(13)
      X(228,NUMPH) = XANIN(1)**4*KA(258)*XMTL(13)
      X(229,NUMPH) = XANIN(1)*KA(259)*XMTL(13)**2
      X(230,NUMPH) = XANIN(2)*KA(260)*XMTL(13)
      X(231,NUMPH) = XANIN(10)*KA(261)*XMTL(13)
      X(232,NUMPH) = XANIN(10)**2*KA(262)*XMTL(13)
      X(233,NUMPH) = XANIN(10)**3*KA(263)*XMTL(13)
      X(234,NUMPH) = XANIN(11)*KA(55)*H*KA(264)*XMTL(13)
      X(235,NUMPH) = XANIN(12)*KA(265)*XMTL(13)
      X(236,NUMPH) = XANIN(4)
      X(237,NUMPH) = XANIN(4)*KA(266)*H
      X(238,NUMPH) = XANIN(4)*KA(266)*KA(267)*H2
      X(239,NUMPH) = XANIN(4)*KA(266)*KA(267)*KA(268)*H3
      X(240,NUMPH) = XANIN(4)*KA(295)*XMTL(1)
      X(241,NUMPH) = XANIN(4)*KA(269)*XMTL(2)
      X(242,NUMPH) = XANIN(4)*KA(270)*XMTL(3)
      X(243,NUMPH) = XANIN(4)**2*KA(271)*XMTL(3)
      X(244,NUMPH) = XANIN(4)*KA(283)*XMTL(4)
      X(245,NUMPH) = XANIN(4)**2*KA(284)*XMTL(4)
      X(246,NUMPH) = XANIN(4)*KA(283)/(KA(285)*H)*XMTL(4)
      X(247,NUMPH) = XANIN(4)*KA(283)/(KA(285)*KA(286)*H2)*XMTL(4)
      X(248,NUMPH) = XANIN(4)*KA(294)*XMTL(6)
      X(249,NUMPH) = XANIN(4)*KA(291)*XMTL(7)
      X(250,NUMPH) = XANIN(4)**2*KA(292)*XMTL(7)
      X(251,NUMPH) = XANIN(4)*KA(291)/(KA(293)*H)*XMTL(7)
      X(252,NUMPH) = XANIN(4)*KA(279)*XMTL(8)
      X(253,NUMPH) = XANIN(4)**2*KA(280)*XMTL(8)
      X(254,NUMPH) = XANIN(4)*XANIN(1)*KA(279)*KA(281)*XMTL(8)
      X(255,NUMPH) = XANIN(4)*KA(272)*XMTL(9)
      X(256,NUMPH) = XANIN(4)**2*KA(273)*XMTL(9)
      X(257,NUMPH) = XANIN(4)*KA(272)*KA(274)*H*XMTL(9)
      X(258,NUMPH) = XANIN(4)*KA(272)/(KA(275)*H)*XMTL(9)
      X(259,NUMPH) = XANIN(4)*KA(296)*XMTL(10)
      X(260,NUMPH) = XANIN(4)**2*KA(297)*XMTL(10)
      X(261,NUMPH) = XANIN(4)*KA(287)*XMTL(11)
      X(262,NUMPH) = XANIN(4)**2*KA(288)*XMTL(11)
      X(263,NUMPH) = XANIN(4)*XANIN(1)*KA(287)*KA(289)*XMTL(11)
      X(264,NUMPH) = XANIN(4)*KA(276)*XMTL(13)
      X(265,NUMPH) = XANIN(4)**2*KA(277)*XMTL(13)
      X(266,NUMPH) = XANIN(4)*KA(276)/(KA(278)*H)*XMTL(13)
      X(268,NUMPH) = XANIN(9)*KA(190)*KA(189)*KA(303)*H3
      X(269,NUMPH) = XANIN(9)*KA(190)*KA(189)*KA(303)*KA(302)*H4
      X(270,NUMPH) = XANIN(14)
      X(271,NUMPH) = BETA*XANIN(14)*KA(304)*10**(SSCDB*PH)*XMTL(7)
      X(272,NUMPH) = BETA*XANIN(14)*KA(305)*XMTL(8)
      X(273,NUMPH) = BETA*XANIN(14)*KA(306)*XMTL(9)
      X(274,NUMPH) = BETA*XANIN(14)*KA(307)*XMTL(4)
      X(275,NUMPH) = BETA*XANIN(14)*KA(308)*XMTL(10)
      X(276,NUMPH) = BETA*XANIN(14)*KA(309)*XMTL(11)
      X(277,NUMPH) = BETA*XANIN(14)*KA(310)*XMTL(12)
      X(278,NUMPH) = BETA*XANIN(14)*KA(311)*XMTL(13)
      X(279,NUMPH) = BETA*XANIN(14)*KA(312)*XMTL(1)
      X(280,NUMPH) = BETA*XANIN(14)*KA(313)*XMTL(2)
      X(281,NUMPH) = BETA*XANIN(14)*KA(314)*XMTL(3)
      X(282,NUMPH) = BETA*XANIN(14)*KA(315)*XMTL(5)
      X(283,NUMPH) = BETA*XANIN(14)*KA(316)*XMTL(6)
      X(284,NUMPH) = XMTL(14)
      X(285,NUMPH) = XANIN(1 )*KA(317)*XMTL(14)
      X(286,NUMPH) = XANIN(1 )**2*KA(318)*XMTL(14)
      X(287,NUMPH) = XANIN(1 )**3*KA(319)*XMTL(14)
      X(288,NUMPH) = XANIN(1 )**4 *KA(320)*XMTL(14)
      X(289,NUMPH) = XANIN(1 )**2*KA(321)*XMTL(14)**2
      X(290,NUMPH) = XANIN(1 )**4*KA(322)*XMTL(14)**3
      X(291,NUMPH) = XANIN(3 )*KA(323)*XMTL(14)
      X(292,NUMPH) = XANIN(3 )*H*KA(324)*XMTL(14)
      X(293,NUMPH) = XANIN(3 )/H*KA(325)*XMTL(14)
      X(294,NUMPH) = XANIN(4 )*KA(326)*XMTL(14)
      X(295,NUMPH) = XANIN(4 )*H*KA(327)*XMTL(14)
      X(296,NUMPH) = XANIN(4 )/H*KA(328)*XMTL(14)
      X(297,NUMPH) = XANIN(4 )/H2*KA(329)*XMTL(14)
      X(298,NUMPH) = XANIN(12)*KA(330)*XMTL(14)
      X(299,NUMPH) = XANIN(5 )*KA(331)*XMTL(14)
      X(300,NUMPH) = XANIN(2 )*KA(332)*XMTL(14)
      X(301,NUMPH) = XANIN(2 )**2*KA(333)*XMTL(14)
      X(302,NUMPH) = XANIN(11)*KA(334)*KA(54)*KA(55)*H2*XMTL(14)
      X(303,NUMPH) = XANIN(11)**2*KA(335)*KA(54)**2*KA(55)**2
     &                   *H**4*XMTL(14)
      X(304,NUMPH) = XANIN(11)**3*KA(336)*KA(54)**3*KA(55)**3
     &                   *H**6*XMTL(14)
      X(305,NUMPH) = XANIN(7 )**2*KA(337)/H2*XMTL(14)
      X(306,NUMPH) = XANIN(15)
      X(307,NUMPH) = XANIN(15)*KA(342)*H
      X(308,NUMPH) = XANIN(15)*KA(343)*XMTL(14)
      X(309,NUMPH) = XANIN(15)**2*KA(344)*XMTL(14)
      X(310,NUMPH) = XANIN(15)**3*KA(345)*XMTL(14)
C      X(311,NUMPH) = XANIN(15)**4*KA(346)*XMTL(14)
      X(312,NUMPH) = XANIN(15)*KA(347)*XMTL(2 )
      X(313,NUMPH) = XANIN(15)*KA(348)*XMTL(3 )
      X(314,NUMPH) = XANIN(15)*KA(349)*XMTL(8 )
      X(315,NUMPH) = XANIN(15)*KA(350)*XMTL(6 )
      X(316,NUMPH) = XANIN(15)*KA(351)*XMTL(11)
      X(317,NUMPH) = XANIN(15)*KA(352)*XMTL(4 )
      X(318,NUMPH) = XANIN(15)**2*KA(353)*XMTL(4 )
      X(319,NUMPH) = XANIN(15)**3*KA(354)*XMTL(4 )
C
C  CALCULATION OF THE TOTAL DISSOLVED VALUES FOR THE ANIONS.
C
      DO 21 I=2,NAVALS
        XA(1,I) = HSP(I)
        XA(2,I) = HSPQ(I)
        XA(3,I) = 0D0
   21 CONTINUE
C
      DO 140 I=1,NMVALS
      IF(XMTL(I).GT.1D-24)THEN
        X1=XMTL(I)
        X2=X1*X1
        DO 22 J=2,NAVALS
          XA(1,J) = XA(1,J) +SPSUM( J,I,1)*X1
          XA(2,J) = XA(2,J) +SPSUM( J,I,2)*X1
          XA(3,J) = XA(3,J) +SPSUM( J,I,3)*X1
   22   CONTINUE
        XA(2,3)=XA(2,3)+SPSUM(3,I,5)*X2
      END IF
  140 CONTINUE
C
      TDSO4 (NUMPH) = XANIN( 2)*
     &(XANIN( 2)*(XANIN( 2)*XA(3,2)*3D0+XA(2,2)*2D0)+(XA(1,2)+1D0))
```

```
      TDEDTA(NUMPH) = XANIN( 3)*
     &(XANIN( 3)*(XANIN( 3)*XA(3,3)*3D0+XA(2,3)*2D0)+(XA(1,3)+1D0))
      TDNTA (NUMPH) = XANIN( 4)*
     &(XANIN( 4)*(XANIN( 4)*XA(3,4)*3D0+XA(2,4)*2D0)+(XA(1,4)+1D0))
      TDIC  (NUMPH) = XANIN( 5)*
     &(XANIN( 5)*(XANIN( 5)*XA(3,5)*3D0+XA(2,5)*2D0)+(XA(1,5)+1D0))
      TDCL  (NUMPH) = XANIN( 6)*
     &(XANIN( 6)*(XANIN( 6)*XA(3,6)*3D0+XA(2,6)*2D0)+(XA(1,6)+1D0))
      TDOC  (NUMPH) = XANIN( 7)*
     &(XANIN( 7)*(XANIN( 7)*XA(3,7)*3D0+XA(2,7)*2D0)+(XA(1,7)+1D0))
      TDSE  (NUMPH) = XANIN( 8)*
     &(XANIN( 8)*(XANIN( 8)*XA(3,8)*3D0+XA(2,8)*2D0)+(XA(1,8)+1D0))
      TDSIO4(NUMPH) = XANIN( 9)*
     &(XANIN( 9)*(XANIN( 9)*XA(3,9)*3D0+XA(2,9)*2D0)+(XA(1,9)+1D0))
      TDNH3 (NUMPH) = XANIN(10)*
     &(XANIN(10)*(XANIN(10)*XA(3,10)*3D0+XA(2,10)*2D0)+(XA(1,10)+1D0))
      TDPO4 (NUMPH) = XANIN(11)*
     &(XANIN(11)*(XANIN(11)*XA(3,11)*3D0+XA(2,11)*2D0)+(XA(1,11)+1D0))
      TDAC  (NUMPH) = XANIN(12)*
     &(XANIN(12)*(XANIN(12)*XA(3,12)*3D0+XA(2,12)*2D0)+(XA(1,12)+1D0))
      TDS   (NUMPH) = XANIN(13)*
     &(XANIN(13)*(XANIN(13)*XA(3,13)*3D0+XA(2,13)*2D0)+(XA(1,13)+1D0))
      TDSS  (NUMPH) = BETA*XANIN(14)*
     &(XANIN(14)*(XANIN(14)*XA(3,14)*3D0+XA(2,14)*2D0)+(XA(1,14)+1D0))
      TDF   (NUMPH) = XANIN(15)*
     &(XANIN(15)*(XANIN(15)*XA(3,15)*3D0+XA(2,15)*2D0)+(XA(1,15)+1D0))
C
C CALCULATION OF THE TOTAL DISSOLVED VALUES OF THE METALS.
C
      DO 23 I=1,NMVALS
        XM(1,I) = SPSUM(1,I,1)
        XM(2,I) = SPSUM(1,I,4)
        XM(3,I) = SPSUM(1,I,7)
        SPSUM(14,I,1)=SPSUM(14,I,1)*BETA
   23 CONTINUE
C
      DO 150 I=2,NAVALS
      IF(XANIN(I).GT.1D-24)THEN
        X1=XANIN(I)
        X2=X1*X1
        X3=X2*X1
        DO 24 J=1,NMVALS
          XM(1,J) = XM(1,J) +SPSUM(I,J,1)*X1 +SPSUM(I,J,2)*X2
     &                      +SPSUM(I,J,3)*X3
          XM(2,J) = XM(2,J) +SPSUM(I,J,4)*X1 +SPSUM(I,J,5)*X2
     &                      +SPSUM(I,J,6)*X3
          XM(3,J) = XM(3,J) +SPSUM(I,J,7)*X1 +SPSUM(I,J,8)*X2
     &                      +SPSUM(I,J,9)*X3
   24   CONTINUE
      END IF
  150 CONTINUE
  944  C
  945        TDNA (NUMPH) = XMTL( 1)*
  946       &(XMTL( 1)*(XMTL( 1)*XM(3,1) *3D0+XM(2,1) *2D0)+(XM(1,1) +1D0))
  947        TDMG (NUMPH) = XMTL( 2)*
  948       &(XMTL( 2)*(XMTL( 2)*XM(3,2) *3D0+XM(2,2) *2D0)+(XM(1,2 )+1D0))
  949        TDCA (NUMPH) = XMTL( 3)*
  950       &(XMTL( 3)*(XMTL( 3)*XM(3,3) *3D0+XM(2,3) *2D0)+(XM(1,3 )+1D0))
  951        TDFE3(NUMPH) = XMTL( 4)*
  952       &(XMTL( 4)*(XMTL( 4)*XM(3,4) *3D0+XM(2,4) *2D0)+(XM(1,4 )+1D0))
  953        TDHG1(NUMPH) = XMTL( 5)*
  954       &(XMTL( 5)*(XMTL( 5)*XM(3,5) *3D0+XM(2,5) *2D0)+(XM(1,5 )+1D0))
  955        TDHG2(NUMPH) = XMTL( 6)*
  956       &(XMTL( 6)*(XMTL( 6)*XM(3,6) *3D0+XM(2,6) *2D0)+(XM(1,6 )+1D0))
  957        TDCD (NUMPH) = XMTL( 7)*
  958       &(XMTL( 7)*(XMTL( 7)*XM(3,7) *3D0+XM(2,7) *2D0)+(XM(1,7 )+1D0))
  959        TDCU (NUMPH) = XMTL( 8)*
  960       &(XMTL( 8)*(XMTL( 8)*XM(3,8) *3D0+XM(2,8) *2D0)+(XM(1,8 )+1D0))
  961        TDFE2(NUMPH) = XMTL( 9)*
  962       &(XMTL( 9)*(XMTL( 9)*XM(3,9) *3D0+XM(2,9) *2D0)+(XM(1,9 )+1D0))
  963        TDMN2(NUMPH) = XMTL(10)*
  964       &(XMTL(10)*(XMTL(10)*XM(3,10)*3D0+XM(2,10)*2D0)+(XM(1,10)+1D0))
  965        TDZN (NUMPH) = XMTL(11)*
  966       &(XMTL(11)*(XMTL(11)*XM(3,11)*3D0+XM(2,11)*2D0)+(XM(1,11)+1D0))
  967        TDK  (NUMPH) = XMTL(12)*
  968       &(XMTL(12)*(XMTL(12)*XM(3,12)*3D0+XM(2,12)*2D0)+(XM(1,12)+1D0))
  969        TDCO (NUMPH) = XMTL(13)*
  970       &(XMTL(13)*(XMTL(13)*XM(3,13)*3D0+XM(2,13)*2D0)+(XM(1,13)+1D0))
  971        TDAL (NUMPH) = XMTL(14)*
  972       &(XMTL(14)*(XMTL(14)*XM(3,14)*3D0+XM(2,14)*2D0)+(XM(1,14)+1D0))
  973  C
  974        RETURN
  975        END
```

```
SPCALC  Local Symbols

Name                          Class   Type              Size   Offset

SSCDB . . . . . . . . . . . . param                             0006
BETA. . . . . . . . . . . . . param                             000a
NAVALS. . . . . . . . . . . . param                             000e
NMVALS. . . . . . . . . . . . param                             0012
NUMPH . . . . . . . . . . . . param                             0016
PH. . . . . . . . . . . . . . param                             001a
XA. . . . . . . . . . . . . . local   REAL*8             360    0000
H . . . . . . . . . . . . . . local   REAL*8               8    00f4
I . . . . . . . . . . . . . . local   INTEGER*4            4    00fc
X1. . . . . . . . . . . . . . local   REAL*8               8    0100
J . . . . . . . . . . . . . . local   INTEGER*4            4    0108
X2. . . . . . . . . . . . . . local   REAL*8               8    010c
X3. . . . . . . . . . . . . . local   REAL*8               8    0114
```

```
SPCALC  Local Symbols

Name                          Class   Type              Size   Offset

H2. . . . . . . . . . . . . . local   REAL*8               8    011c
H3. . . . . . . . . . . . . . local   REAL*8               8    0124
H4. . . . . . . . . . . . . . local   REAL*8               8    012c
XM. . . . . . . . . . . . . . local   REAL*8             336    0168
TDHG1 . . . . . . . . . . . . OUT1    REAL*4             164    0b88
TDHG2 . . . . . . . . . . . . OUT1    REAL*4             164    0c2c
TDCD. . . . . . . . . . . . . OUT1    REAL*4             164    0cd0
TDCU. . . . . . . . . . . . . OUT1    REAL*4             164    0d74
TDFE2 . . . . . . . . . . . . OUT1    REAL*4             164    0e18
TDMN2 . . . . . . . . . . . . OUT1    REAL*4             164    0ebc
TDZN. . . . . . . . . . . . . OUT1    REAL*4             164    0f60
TDK . . . . . . . . . . . . . OUT1    REAL*4             164    1004
TDCO. . . . . . . . . . . . . OUT1    REAL*4             164    10a8
TDSS. . . . . . . . . . . . . OUT1    REAL*4             164    07b0
TDAL. . . . . . . . . . . . . OUT1    REAL*4             164    114c
TDF . . . . . . . . . . . . . OUT1    REAL*4             164    0854
C . . . . . . . . . . . . . . OUT2    REAL*8            2848    0000
M . . . . . . . . . . . . . . OUT2    REAL*8            2848    0b20
ALPHA . . . . . . . . . . . . OUT2    REAL*8            2848    1640
KA. . . . . . . . . . . . . . OUT2    REAL*8            2848    2160
GAM . . . . . . . . . . . . . OUT2    REAL*8            2552    2c80
KTAG. . . . . . . . . . . . . OUT2    CHAR*2             712    3678
SPSUM . . . . . . . . . . . . CALC    REAL*8           15120    0000
HSP . . . . . . . . . . . . . CALC    REAL*8             120    3b10
HSPQ. . . . . . . . . . . . . CALC    REAL*8             120    3b88
XANIN . . . . . . . . . . . . CALC    REAL*8             120    3c00
XMTL. . . . . . . . . . . . . CALC    REAL*8             112    3c78
XPREV . . . . . . . . . . . . CALC    REAL*8             112    3ce8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
TDSO4 . . . . . . . . . . . . OUT1    REAL*4             164    0000
TDEDTA. . . . . . . . . . . . OUT1    REAL*4             164    00a4
TDNTA . . . . . . . . . . . . OUT1    REAL*4             164    0148
TDIC. . . . . . . . . . . . . OUT1    REAL*4             164    01ec
TDCL. . . . . . . . . . . . . OUT1    REAL*4             164    0290
TDOC. . . . . . . . . . . . . OUT1    REAL*4             164    0334
TDSE. . . . . . . . . . . . . OUT1    REAL*4             164    03d8
TDSIO4. . . . . . . . . . . . OUT1    REAL*4             164    047c
TDNH3 . . . . . . . . . . . . OUT1    REAL*4             164    0520
TDPO4 . . . . . . . . . . . . OUT1    REAL*4             164    05c4
TDAC. . . . . . . . . . . . . OUT1    REAL*4             164    0668
TDS . . . . . . . . . . . . . OUT1    REAL*4             164    070c
TDNA. . . . . . . . . . . . . OUT1    REAL*4             164    08f8
TDMG. . . . . . . . . . . . . OUT1    REAL*4             164    099c
TDCA. . . . . . . . . . . . . OUT1    REAL*4             164    0a40
TDFE3 . . . . . . . . . . . . OUT1    REAL*4             164    0ae4
```

```
C
C
      SUBROUTINE OUTPUT(NAVALS,NMVALS,A,MTL)
C     -----------------
C
C     THIS ROUTINE PRINTS THE RESULTS OF THE PROGRAM.  THE OUTPUT
C     SEQUENCE IS AS FOLLOWS, WHERE EACH REPORT IS OPTIONAL AND
C     (2)-(3) MAY BE PRINTED IN ANY OR ALL OF THE MEASUREMENT UNITS:
C
C       (1) EQUILIBRIUM STABILITY CONSTANTS DEFINITIONS REPORT
C       (2) ANION SPECIES REPORT
C       (3) METAL SPECIES REPORT
C
      INTEGER NUMPH,NAVALS,NMVALS,PR,NPL
      REAL*8 X(319,41)
      REAL*4 TDSO4(41),TDEDTA(41),TDNTA(41),TDIC(41),TDCL(41),TDOC(41),
     &TDSE(41),TDSIO4(41),TDNH3(41),TDPO4(41),TDAC(41),TDS(41),TDNA(41)
     &TDMG(41),TDCA(41),TDFE3(41),TDHG1(41),TDHG2(41),TDCD(41),TDCU(41)
     &TDFE2(41),TDMN2(41),TDZN(41),TDK(41),TDCO(41),TDSS(41),TDF(41),
     &TDAL(41),PHS(41)
      REAL*8 A(15),MTL(14)
      REAL*8 CONVTA(15),CONVTM(14)
      INTEGER ANION,SOLID,ASLD(15),MTLSLD(14),S(63),J
      LOGICAL*2 S$DEFNS,S$ASPEC(15),S$MSPEC(14)
      CHARACTER*27 SLDS(63),SLDID
      INTEGER UNITS(6)
      CHARACTER*15 AFILES(15),MTLFILES(14),FILE
      CHARACTER*4 FILEID
      COMMON /BOTH/CONVTA,CONVTM,ASLD,MTLSLD,UNITS,
     &             FILEID,AFILES,MTLFILES,S$DEFNS,S$ASPEC,S$MSPEC
      COMMON /OUT/X,PHS
      COMMON /OUT1/TDSO4,TDEDTA,TDNTA,TDIC,TDCL,TDOC,
     &            TDSE,TDSIO4,TDNH3,TDPO4,TDAC,TDS,TDSS,TDF,
     &            TDNA,TDMG,TDCA,TDFE3,TDHG1,TDHG2,TDCD,TDCU,
     &            TDFE2,TDMN2,TDZN,TDK,TDCO,TDAL
      REAL*8 C(356),M(356),ALPHA(356),KA(356),GAM(319)
      CHARACTER KTAG(356)*2
      CHARACTER FF
      COMMON /OUT2/C,M,ALPHA,KA,GAM,KTAG
      REAL*8 F,F2,F3
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
      DATA FF/12/
      DATA NPL/20/
      DATA (SLDS(I),I=1,10)
     &/'Cd(OH)2                    ','CuO - TENORITE             ',
     & 'Cu(OH)2                    ','Cu4(OH)6SO4 - BROCHANTITE  ',
     & 'MgCa(CO3)2 - DOLOMITE      ','CaCO3 - CALCITE            ',
     & 'CdCO3                      ','CuCO3                      ',
     & 'Cu2(OH)2CO3 - MALACHITE    ','Cu3(OH)2(CO3)2 - AZURITE   '/
      DATA (SLDS(I),I=11,20)
     &/'Cu2(OH)3Cl2 - ATACAMITE    ','Mg5(OH)PO4                 ',
     & 'MgHPO4                     ','Mg3(PO4)2(8H2O)            ',
     & 'Mg4H(PO4)3                 ','Ca5OH(PO4)3                ',
     & 'CaHPO4                     ','Ca3(PO4)2                  ',
     & 'Ca4H(PO4)3                 ','Cd3(PO4)2                  '/
      DATA (SLDS(I),I=21,30)
     &/'Cu3(PO4)2                  ','CdS                        ',
     & 'CuS - COVELLITE            ','FeS - MACKINAWITE          ',
     & 'Fe(OH)2                    ','Fe3(PO4)2(8H2O) - VIVIANITE',
     & 'FeCO3 - SIDERITE          ','HgO - MONTROYDITE        ',
     & 'HgS(red) - CINNABAR        ','HgS(black) - METACINNABAR  '/
      DATA (SLDS(I),I=31,40)
     &/'HgSeO3                     ','CuSeO3                     ',
     & 'MgSeO3                     ','Mg2(HSiO4)3                ',
     & 'Mn(OH)2                    ','MnCO3                      ',
     & 'MnS(pink)                  ','MnS(green)                 ',
     & 'MnSeO3                     ','Zn(OH)2                    '/
      DATA (SLDS(I),I=41,50)
     &/'ZnO                        ','ZnCO3                      ',
     & 'Zn3(PO4)2(4H2O)            ','ZnS                        ',
     & 'CaSiO4                     ','Hg2CO3                     ',
     & 'Fe(PO4)(2H2O)              ','Hg2HPO4                    ',
     & 'Hg2SO4                     ','Hg2Cl2                     '/
      DATA (SLDS(I),I=51,60)
     &/'FeOOH                      ','Fe(OH)3                    ',
     & 'Fe2O3                      ','CoCO3                      ',
     & 'Co(OH)2                    ','Co3(PO4)2                  ',
     & 'CoS                        ','Al(OH)3 - GIBBSITE         ',
     & 'Al(OH)3 - AMORPHOUS        ','AlPO4                      '/
      DATA (SLDS(I),I=61,63)
     &/'AlH2PO4(OH)2               ','MgF2                       ',
     & 'CaF2                       '/
C
      DATA S/ 40, 20, 30, 69,  7,  6, 62, 39, 65, 66, 70, 82,
     &        83, 84, 85, 75, 76, 77, 78, 68, 67, 61, 63, 94,
     &        95, 96, 97,120,122,121,129,130,131,200,201,202,
     &       203,204,205,206,207,208,209,210,211,216,217,218,
     &       219,212,214,213,215,298,299,300,301,338,339,340,
     &       341,355,356/

C
C  OUTPUT OF DEFINITIONS TABLES.
C
C
      IF(S$DEFNS) CALL WDEFNS(PR,SLDS,S)
C
C
C  OUTPUT OF ANION SPECIES TABLES.
C
C
      DO 1 ANION=2,NAVALS
      IF(S$ASPEC(ANION))THEN
        SOLID=ASLD(ANION)
        IF(SOLID.EQ.0)THEN
          SLDID=' '
        ELSE
          SLDID=SLDS(SOLID)
        END IF
! DO CASE ANION
      F=1.
      F2=1.
      F3=1.
      IF (ANION.EQ.1) THEN
! HYDROXIDE
      ELSE IF (ANION.EQ.2) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2
            F3=3
          END IF
          FILE=AFILES(2)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WSO4(TDSO4,A(2),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.3) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
          END IF
          FILE=AFILES(3)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WEDTA(TDEDTA,A(3),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.4) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
          END IF
          FILE=AFILES(4)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WNTA(TDNTA,A(4),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.5) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4)THEN
```

```
            F=CONVTA(ANION)
            F2=2.
          END IF
          FILE=AFILES(5)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WC(TDIC,A(5),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.6) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
            F3=3.
          END IF
          FILE=AFILES(6)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WCL(TDCL,A(6),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.7) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
          END IF
          FILE=AFILES(7)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WOC(TDOC,A(7),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.8) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
          END IF
          FILE=AFILES(8)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WSE(TDSE,A(8),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.9) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
          END IF
          FILE=AFILES(9)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WSIO4(TDSIO4,A(9),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.10) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
            F3=3.
          END IF
          FILE=AFILES(10)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WNH3(TDNH3,A(10),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.11) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
          END IF
          FILE=AFILES(11)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WPO4(TDPO4,A(11),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.12) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
            F3=3.
          END IF
          FILE=AFILES(12)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WAC(TDAC,A(12),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.13) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
            F3=3.
          END IF
          FILE=AFILES(13)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WS(TDS,A(13),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (ANION.EQ.14) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF (UNITS(I).LE.3) THEN
            J=1
            DO WHILE (J .LE. NUMPH)
              TDSS(J)=0D0
              DO K=271,283
                TDSS(J)=TDSS(J)+X(K,J)
              END DO
              J=J+1
            END DO
            FILE=AFILES(14)(1:8)//'.DA'//CHAR(UNITS(I)+48)
            CALL WSS(TDSS,A(14),UNITS(I),FILE)
          ELSE
            F=1.
            F2=2.
            F3=3.
            DO 10 J=1,NUMPH
              X(271,J) = X(271,J)*CONVTM(7)
              X(272,J) = X(272,J)*CONVTM(8)
              X(273,J) = X(273,J)*CONVTM(9)
              X(274,J) = X(274,J)*CONVTM(3)
              X(275,J) = X(275,J)*CONVTM(10)
              X(276,J) = X(276,J)*CONVTM(11)
              X(277,J) = X(277,J)*CONVTM(12)
              X(278,J) = X(278,J)*CONVTM(13)
              X(279,J) = X(279,J)*CONVTM(1)
              X(280,J) = X(280,J)*CONVTM(2)
              X(281,J) = X(281,J)*CONVTM(3)
              X(282,J) = X(282,J)*CONVTM(5)
              X(283,J) = X(283,J)*CONVTM(6)
              TDSS(J)=0D0
              DO K=271,283
                TDSS(J)=TDSS(J) +X(K,J)
              END DO
   10       CONTINUE
            FILE=AFILES(14)(1:8)//'.DA'//CHAR(UNITS(I)+48)
            CALL WSS(TDSS,A(14),UNITS(I),FILE)
            DO 11 J=1,NUMPH
              X(271,J) = X(271,J)/CONVTM(7)
              X(272,J) = X(272,J)/CONVTM(8)
              X(273,J) = X(273,J)/CONVTM(9)
              X(274,J) = X(274,J)/CONVTM(3)
              X(275,J) = X(275,J)/CONVTM(10)
              X(276,J) = X(276,J)/CONVTM(11)
              X(277,J) = X(277,J)/CONVTM(12)
              X(278,J) = X(278,J)/CONVTM(13)
              X(279,J) = X(279,J)/CONVTM(1)
              X(280,J) = X(280,J)/CONVTM(2)
              X(281,J) = X(281,J)/CONVTM(3)
              X(282,J) = X(282,J)/CONVTM(5)
              X(283,J) = X(283,J)/CONVTM(6)
   11       CONTINUE
          END IF
          I=I+1
```

```
        END DO
      ELSE IF (ANION.EQ.15) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTA(ANION)
            F2=2.
            F3=3.
          END IF
          FILE=AFILES(15)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WF(TDF,A(15),UNITS(I),FILE)
          I=I+1
        END DO
      END IF
      END IF
    1 CONTINUE
C
C
C  OUTPUT OF METAL SPECIES TABLES.
C
      DO 2 METAL=1,NMVALS
      IF(S$MSPEC(METAL))THEN
        SOLID=MTLSLD(METAL)
        IF(SOLID.EQ.0)THEN
          SLDID=' '
        ELSE
          SLDID=SLDS(SOLID)
        END IF
      F=1.
      F1=1.
      F2=1.
!  DO CASE METAL
      IF (METAL.EQ.1) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
          END IF
          FILE=MTLFILES(1)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WNA(TDNA,MTL(1),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.2) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
          END IF
          FILE=MTLFILES(2)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WMG(TDMG,MTL(2),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.3) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
          END IF
          FILE=MTLFILES(3)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WCA(TDCA,MTL(3),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.4) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
            F2=2.
            F3=3.
          END IF
          FILE=MTLFILES(4)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WFEIII(TDFE3,MTL(4),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.5) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
          END IF
          FILE=MTLFILES(5)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WHGI(TDHG1,MTL(5),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.6) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
            F3=3.
          END IF
          FILE=MTLFILES(6)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WHGII(TDHG2,MTL(6),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.7) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
            F2=2.
          END IF
          FILE=MTLFILES(7)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WCD(TDCD,MTL(7),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.8) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
            F2=2.
          END IF
          FILE=MTLFILES(8)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WCU(TDCU,MTL(8),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.9) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
          END IF
          FILE=MTLFILES(9)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WFEII(TDFE2,MTL(9),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.10) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
            F2=2.
          END IF
          FILE=MTLFILES(10)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WMNII(TDMN2,MTL(10),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.11) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
            F=CONVTM(METAL)
            F2=2.
          END IF
          FILE=MTLFILES(11)(1:8)//'.DA'//CHAR(UNITS(I)+48)
          CALL WZN(TDZN,MTL(11),UNITS(I),FILE)
          I=I+1
        END DO
      ELSE IF (METAL.EQ.12) THEN
        I=1
        DO WHILE (UNITS(I).NE.0)
          IF(UNITS(I).GE.4) THEN
```

```
  1434              F=CONVTM(METAL)
  1435            END IF
  1436            FILE=MTLFILES(12)(1:8)//'.DA'//CHAR(UNITS(I)+48)
  1437            CALL WK(TDK,MTL(12),UNITS(I),FILE)
  1438            I=I+1
  1439          END DO
  1440        ELSE IF (METAL.EQ.13) THEN
  1441          I=1
  1442          DO WHILE (UNITS(I).NE.0)
  1443            IF(UNITS(I).GE.4) THEN
  1444              F=CONVTM(METAL)
  1445              F2=2.
  1446            END IF
  1447            FILE=MTLFILES(13)(1:8)//'.DA'//CHAR(UNITS(I)+48)
  1448            CALL WCO(TDCO,MTL(13),UNITS(I),FILE)
  1449            I=I+1
  1450          END DO
  1451        ELSE IF (METAL.EQ.14) THEN
  1452          I=1
  1453          DO WHILE (UNITS(I).NE.0)
  1454            IF(UNITS(I).GE.4) THEN
  1455              F=CONVTM(METAL)
  1456              F2=2.
  1457                  F3=3.
  1458            END IF
  1459            FILE=MTLFILES(14)(1:8)//'.DA'//CHAR(UNITS(I)+48)
  1460            CALL WAL(TDAL,MTL(14),UNITS(I),FILE)
  1461            I=I+1
  1462          END DO
  1463        END IF
  1464        END IF
  1465      2 CONTINUE
  1466  C
  1467        WRITE(PR,20) FF
  1468     20 FORMAT(A)
  1469        RETURN
  1470        END
```

```
OUTPUT  Local Symbols

Name                        Class   Type             Size   Offset

MTL . . . . . . . . . . . . param                            0006
A . . . . . . . . . . . . . param                            000a
NMVALS. . . . . . . . . . . param                            000e
NAVALS. . . . . . . . . . . param                            0012
I . . . . . . . . . . . . . local   INTEGER*4           4    0136
J . . . . . . . . . . . . . local   INTEGER*4           4    013a
K . . . . . . . . . . . . . local   INTEGER*4           4    013e
FILE. . . . . . . . . . . . local   CHAR*15            15    0142
```

```
OUTPUT  Local Symbols

Name                          Class   Type              Size   Offset

METAL . . . . . . . . . . . . local   INTEGER*4            4   0152
ANION . . . . . . . . . . . . local   INTEGER*4            4   0156
F1. . . . . . . . . . . . . . local   REAL*4               4   015a
SOLID . . . . . . . . . . . . local   INTEGER*4            4   015e
FF. . . . . . . . . . . . . . local   CHAR*1               1   0311
SLDS. . . . . . . . . . . . . local   CHAR*27           1701   0312
S . . . . . . . . . . . . . . local   INTEGER*4          252   09b8
TDSO4 . . . . . . . . . . . . OUT1    REAL*4             164   0000
TDEDTA. . . . . . . . . . . . OUT1    REAL*4             164   00a4
TDNTA . . . . . . . . . . . . OUT1    REAL*4             164   0148
TDIC. . . . . . . . . . . . . OUT1    REAL*4             164   01ec
TDCL. . . . . . . . . . . . . OUT1    REAL*4             164   0290
TDOC. . . . . . . . . . . . . OUT1    REAL*4             164   0334
TDSE. . . . . . . . . . . . . OUT1    REAL*4             164   03d8
TDSIO4. . . . . . . . . . . . OUT1    REAL*4             164   047c
TDNH3 . . . . . . . . . . . . OUT1    REAL*4             164   0520
TDPO4 . . . . . . . . . . . . OUT1    REAL*4             164   05c4
TDAC. . . . . . . . . . . . . OUT1    REAL*4             164   0668
TDS . . . . . . . . . . . . . OUT1    REAL*4             164   070c
TDNA. . . . . . . . . . . . . OUT1    REAL*4             164   08f8
TDMG. . . . . . . . . . . . . OUT1    REAL*4             164   099c
TDCA. . . . . . . . . . . . . OUT1    REAL*4             164   0a40
TDFE3 . . . . . . . . . . . . OUT1    REAL*4             164   0ae4
TDHG1 . . . . . . . . . . . . OUT1    REAL*4             164   0b88
TDHG2 . . . . . . . . . . . . OUT1    REAL*4             164   0c2c
TDCD. . . . . . . . . . . . . OUT1    REAL*4             164   0cd0
TDCU. . . . . . . . . . . . . OUT1    REAL*4             164   0d74
TDFE2 . . . . . . . . . . . . OUT1    REAL*4             164   0e18
TDMN2 . . . . . . . . . . . . OUT1    REAL*4             164   0ebc
TDZN. . . . . . . . . . . . . OUT1    REAL*4             164   0f60
TDK . . . . . . . . . . . . . OUT1    REAL*4             164   1004
TDCO. . . . . . . . . . . . . OUT1    REAL*4             164   10a8
TDSS. . . . . . . . . . . . . OUT1    REAL*4             164   07b0
TDF . . . . . . . . . . . . . OUT1    REAL*4             164   0854
TDAL. . . . . . . . . . . . . OUT1    REAL*4             164   114c
PHS . . . . . . . . . . . . . OUT     REAL*4             164   98b8
CONVTA. . . . . . . . . . . . BOTH    REAL*8             120   0000
CONVTM. . . . . . . . . . . . BOTH    REAL*8             112   0078
ASLD. . . . . . . . . . . . . BOTH    INTEGER*4           60   00e8
MTLSLD. . . . . . . . . . . . BOTH    INTEGER*4           56   0124
S$DEFNS . . . . . . . . . . . BOTH    LOGICAL*2            2   032c
S$ASPEC . . . . . . . . . . . BOTH    LOGICAL*2           30   032e
S$MSPEC . . . . . . . . . . . BOTH    LOGICAL*2           28   034c
SLDID . . . . . . . . . . . . W       CHAR*27             27   0020
UNITS . . . . . . . . . . . . BOTH    INTEGER*4           24   015c
AFILES. . . . . . . . . . . . BOTH    CHAR*15            225   0178
```

```
OUTPUT  Local Symbols

Name                        Class   Type              Size   Offset

MTLFILES. . . . . . . . . . BOTH    CHAR*15            210    0259
FILEID. . . . . . . . . . . BOTH    CHAR*4               4    0174
C . . . . . . . . . . . . . OUT2    REAL*8            2848    0000
M . . . . . . . . . . . . . OUT2    REAL*8            2848    0b20
ALPHA . . . . . . . . . . . OUT2    REAL*8            2848    1640
KA. . . . . . . . . . . . . OUT2    REAL*8            2848    2160
GAM . . . . . . . . . . . . OUT2    REAL*8            2552    2c80
KTAG. . . . . . . . . . . . OUT2    CHAR*2             712    3678
F . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . W       INTEGER*4            4    0018
PR. . . . . . . . . . . . . W       INTEGER*4            4    003c
NPL . . . . . . . . . . . . W       INTEGER*4            4    001c
X . . . . . . . . . . . . . OUT     REAL*8          104632    0000

  1471  C
  1472        SUBROUTINE WDEFNS(PR,SLDS,S)
  1473  C     -------------------
  1474  C
  1475  C     THIS BLOCK PRINTS THE VALUES OF THE EQUILIBRIUM
  1476  C     STABILITY CONSTANTS 'K', 'GAM' AND 'ALPHA'.
  1477  C
  1478        INTEGER I,PR
  1479  C
  1480        INTEGER S(63)
  1481        CHARACTER*27 SLDS(63)
  1482        REAL*8 C(356),M(356),ALPHA(356),KA(356),GAM(319)
  1483        CHARACTER*2 KTAG(356)
  1484        COMMON /OUT2/C,M,ALPHA,KA,GAM,KTAG
  1485        CHARACTER FF
  1486        DATA FF/12/
  1487  C
  1488        CALL WHEAD
  1489        WRITE(PR,90) (KTAG(I),I,KA(I)*ALPHA(I),I=1,110)
  1490        CALL WFOOT(2)
  1491  C
  1492        CALL WHEAD
  1493        WRITE(PR,90) (KTAG(I),I,KA(I)*ALPHA(I),I=111,220)
  1494        CALL WFOOT(2)
  1495  C
  1496        CALL WHEAD
  1497        WRITE(PR,90) (KTAG(I),I,KA(I)*ALPHA(I),I=221,303)
  1498        WRITE(PR,91) (KTAG(I),I,KA(I)*ALPHA(I),I=304,316)
  1499        CALL WFOOT(6)
  1500  C
```

```
      CALL WHEAD
      WRITE(PR,90) (KTAG(I),I,KA(I)*ALPHA(I),I=317,356)
      WRITE(PR,98) '* INDICATES K VALUE CORRECTED FOR SPECIFIED ',
     &       'TEMPERATURE.  OTHERS AT 25C.'
      CALL WFOOT(27)
C
      CALL WHEAD
      WRITE(PR,92) 'GAMMA VALUES',('GAMMA',I,GAM(I),I=1,110)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,92) 'GAMMA VALUES',('GAMMA',I,GAM(I),I=111,220)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,92) 'GAMMA VALUES',('GAMMA',I,GAM(I),I=221,319)
      CALL WFOOT(6)
C
      CALL WHEAD
      WRITE(PR,92) 'ALPHA VALUES',('ALPHA',I,ALPHA(I),I=1,110)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,92) 'ALPHA VALUES',('ALPHA',I,ALPHA(I),I=111,220)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,92) 'ALPHA VALUES',('ALPHA',I,ALPHA(I),I=221,330)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,92) 'ALPHA VALUES',('ALPHA',I,ALPHA(I),I=331,356)
      CALL WFOOT(34)
C
      CALL WHEAD
      WRITE(PR,93) 'C VALUES',('C',I,C(I),I=1,110)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,93) 'C VALUES',('C',I,C(I),I=111,220)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,93) 'C VALUES',('C',I,C(I),I=221,330)
      CALL WFOOT(2)
C
      CALL WHEAD
      WRITE(PR,93) 'C VALUES',('C',I,C(I),I=331,356)
      CALL WFOOT(34)
C
      CALL WHEAD
```

```
 1552         WRITE(PR,94) (I,SLDS(I),S(I),KA(S(I))*ALPHA(S(I)),
 1553        & S(I),C(S(I)),S(I),M(S(I)),I=1,40)
 1554         CALL WFOOT(6)
 1555   C
 1556         CALL WHEAD
 1557         WRITE(PR,94) (I,SLDS(I),S(I),KA(S(I))*ALPHA(S(I)),
 1558        & S(I),C(S(I)),S(I),M(S(I)),I=41,63)
 1559         CALL WFOOT(23)
 1560   C
 1561      90 FORMAT(/T67,'K VALUES'/
 1562        &       (/4X,5(3X,A2,'K(',I3,') =',1P,D12.5)))
 1563      91 FORMAT((/4X,5(2X,A2,'Kd(',I3,') =',1P,D12.5)))
 1564      98 FORMAT(//6X,A44,A28)
 1565      92 FORMAT(/T65,A12/(/7X,5(4X,A5,'(',I3,') =',F8.5)))
 1566      93 FORMAT(/T67,A8/(/7X,5(4X,A1,'(',I3,') =',F12.6)))
 1567      94 FORMAT(60X,'SOLIDS TABLE'//
 1568        &       (' ',21X,I2,A29,'   K(',I3,')=',1P,D10.4,0P,
 1569        &         '   C(',I3,')=',F9.4,'   M(',I3,')=',F10.3))
 1570   C
 1571         RETURN
 1572         END


WDEFNS  Local Symbols

Name                         Class   Type              Size   Offset

S . . . . . . . . . . . . . .param                             0006
SLDS. . . . . . . . . . . . .param                             000a
PR. . . . . . . . . . . . . .param                             000e
I . . . . . . . . . . . . . .local   INTEGER*4            4    0162
FF. . . . . . . . . . . . . .local   CHAR*1               1    0ba5
KA. . . . . . . . . . . . . .OUT2    REAL*8            2848    2160
GAM . . . . . . . . . . . . .OUT2    REAL*8            2552    2c80
KTAG. . . . . . . . . . . . .OUT2    CHAR*2             712    3678
C . . . . . . . . . . . . . .OUT2    REAL*8            2848    0000
M . . . . . . . . . . . . . .OUT2    REAL*8            2848    0b20
ALPHA . . . . . . . . . . . .OUT2    REAL*8            2848    1640

 1573   C
 1574         SUBROUTINE PPS(TITLE)
 1575   C     ------------------------
 1576         INTEGER PR/6/,PAGE,SKIP
 1577         CHARACTER TITLE*48
 1578         CHARACTER FF,LF
 1579         INTEGER*2 IYEAR,IMONTH,IDAY
 1580
 1581         DATA FF/12/,LF/10/
 1582
 1583          CALL GETDAT(IYEAR,IMONTH,IDAY)
 1584          PAGE=0
 1585  C
 1586          ENTRY WHEAD
 1587  C
 1588  C       THIS ROUTINE PRINTS THE PAGE HEADINGS AT THE TOP OF A NEW PAGE
 1589  C
 1590          PAGE=PAGE+1
 1591          WRITE(PR,3) FF,IDAY,IMONTH,IYEAR,PAGE
 1592          WRITE(PR,4) TITLE
 1593          RETURN
 1594  C
 1595          ENTRY WFOOT(SKIP)
 1596  C
 1597  C       THIS ROUTINE PRINTS A FOOTNOTE AND PAGE NUMBER AT THE BOTTOM
 1598  C       OF EACH PAGE.
 1599  C
 1600          DO 1 I=1,SKIP
 1601        1 WRITE(PR,2)
 1602          WRITE(PR,5)
 1603          WRITE(PR,3) LF,IDAY,IMONTH,IYEAR,PAGE
 1604          RETURN
 1605  C
 1606        2 FORMAT(' ')
 1607        3 FORMAT(A1,T6,I2,'/',I2,'/',I4,T126,'PAGE',I3)
 1608        4 FORMAT(/T46,'DEPARTMENT OF FISHERIES & OCEANS - WESTERN REGION'
 1609         & /,T53,'CHEMICAL METHODOLOGY - R. WAGEMANN'//T46,A48)
 1610        5 FORMAT(T6,'NOTE : ''T.D.'' MEANS TOTAL DISSOLVED.'/
 1611         & ' ',T6,'NOTE : ALL MOLES/L AND MG/L VALUES ARE ',
 1612         & 'CONCENTRATIONS AS OPPOSED TO ACTIVITIES.'/
 1613         & ' ',T6,'NOTE : A CONCENTRATION OF 0 CAN MEAN EITHER ',
 1614         & 'THE T.D. CONCENTRATION WAS 0 OR THE SPECIES ',
 1615         & 'CONCENTRATION WAS APPROACHING 0.')
 1616  C
 1617          END
```

```
PPS  Local Symbols

Name                          Class   Type              Size   Offset

TITLE . . . . . . . . . . . . local   far pointer          4    0166
SKIP. . . . . . . . . . . . . local   far pointer          4    016a
I . . . . . . . . . . . . . . local   INTEGER*4            4    016e
IMONTH. . . . . . . . . . . . local   INTEGER*2            2    0172
PAGE. . . . . . . . . . . . . local   INTEGER*4            4    0174
IDAY. . . . . . . . . . . . . local   INTEGER*2            2    0178
IYEAR . . . . . . . . . . . . local   INTEGER*2            2    017a
PR. . . . . . . . . . . . . . local   INTEGER*4            4    0cf2
FF. . . . . . . . . . . . . . local   CHAR*1               1    0cf6
LF. . . . . . . . . . . . . . local   CHAR*1               1    0cf7
```

```
Global Symbols

Name                      Class   Type              Size   Offset

BOTH. . . . . . . . . . . common  ***                872     0000
CALC. . . . . . . . . . . common  ***              15704     0000
GETDAT. . . . . . . . . . extern  ***                ***      ***
NWANIN. . . . . . . . . . FSUBRT  ***                ***     10cd
NWMTLS. . . . . . . . . . FSUBRT  ***                ***     0000
OUT . . . . . . . . . . . common  ***             104796     0000
OUT1. . . . . . . . . . . common  ***               4592     0000
OUT2. . . . . . . . . . . common  ***              14656     0000
OUTPUT. . . . . . . . . . FSUBRT  ***                ***     a4c8
PPS . . . . . . . . . . . FSUBRT  ***                ***     dbea
ROOTS . . . . . . . . . . extern  ***                ***      ***
SPCALC. . . . . . . . . . FSUBRT  ***                ***     1efa
TERM. . . . . . . . . . . extern  REAL*8             ***      ***
W . . . . . . . . . . . . common  ***                 64     0000
WAC . . . . . . . . . . . extern  ***                ***      ***
WAL . . . . . . . . . . . extern  ***                ***      ***
WC. . . . . . . . . . . . extern  ***                ***      ***
WCA . . . . . . . . . . . extern  ***                ***      ***
WCD . . . . . . . . . . . extern  ***                ***      ***
WCL . . . . . . . . . . . extern  ***                ***      ***
WCO . . . . . . . . . . . extern  ***                ***      ***
WCU . . . . . . . . . . . extern  ***                ***      ***
WDEFNS. . . . . . . . . . FSUBRT  ***                ***     d020
WEDTA . . . . . . . . . . extern  ***                ***      ***
WF. . . . . . . . . . . . extern  ***                ***      ***
WFEII . . . . . . . . . . extern  ***                ***      ***
WFEIII. . . . . . . . . . extern  ***                ***      ***
WFOOT . . . . . . . . . . FSUBRT  ***                ***     dc74
WHEAD . . . . . . . . . . FSUBRT  ***                ***     dc67
WHGI. . . . . . . . . . . extern  ***                ***      ***
WHGII . . . . . . . . . . extern  ***                ***      ***
WK. . . . . . . . . . . . extern  ***                ***      ***
WMG . . . . . . . . . . . extern  ***                ***      ***
WMNII . . . . . . . . . . extern  ***                ***      ***
WNA . . . . . . . . . . . extern  ***                ***      ***
WNH3. . . . . . . . . . . extern  ***                ***      ***
WNTA. . . . . . . . . . . extern  ***                ***      ***
WOC . . . . . . . . . . . extern  ***                ***      ***
WPO4. . . . . . . . . . . extern  ***                ***      ***
WS. . . . . . . . . . . . extern  ***                ***      ***
WSE . . . . . . . . . . . extern  ***                ***      ***
WSIO4 . . . . . . . . . . extern  ***                ***      ***
WSO4. . . . . . . . . . . extern  ***                ***      ***
WSS . . . . . . . . . . . extern  ***                ***      ***
WZN . . . . . . . . . . . extern  ***                ***      ***
```

```
Code size = dd45 (56645)
Data size = 0dac (3500)
Bss size  = 017c (380)

No errors detected
```

```
      SUBROUTINE WEDTA(TDEDTA,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE EDTA SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDEDTA(41)
      REAL*8 X(319,41),Y(39)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MINO(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CARBON               '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CARBON          '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CARBON     '
        WRITE(PR,203) LABEL,SLDID
        WRITE(PR,114)
        DO 14 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6021) PHS(I),
     &      LOGFN(X(189,I)*F),LOGFN(X(190,I)*F),
     &      LOGFN(X(191,I)*F),LOGFN(X(192,I)*F),
     &      LOGFN(X(193,I)*F),LOGFN(X(194,I)*F),
     &      LOGFN(X(195,I)*F),LOGFN(X(196,I)*F),
     &      LOGFN(X(197,I)*F),LOGFN(X(198,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6021) PHS(I),
     &      X(189,I)/T,X(190,I)/T,
     &      X(191,I)/T,X(192,I)/T,
     &      X(193,I)/T,X(194,I)/T,
     &      X(195,I)/T,X(196,I)/T,
     &      X(197,I)/T,X(198,I)/T
          ELSE
            WRITE(PR,602) PHS(I),
     &      X(189,I)*F,X(190,I)*F,
     &      X(191,I)*F,X(192,I)*F,
     &      X(193,I)*F,X(194,I)*F,
     &      X(195,I)*F,X(196,I)*F,
     &      X(197,I)*F,X(198,I)*F
          END IF
   14   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,203) LABEL,SLDID
        WRITE(PR,115)
        DO 15 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5811) PHS(I),
     &      LOGFN(X(199,I)*F)   ,LOGFN(X(200,I)*F),
     &      LOGFN(X(201,I)*F)   ,LOGFN(X(202,I)*F),
     &      LOGFN(X(203,I)*F*F2),LOGFN(X(204,I)*F),
     &      LOGFN(X(205,I)*F)   ,LOGFN(X(206,I)*F),
     &      LOGFN(X(207,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,5811) PHS(I),
     &      X(199,I)/T   ,X(200,I)/T,
     &      X(201,I)/T   ,X(202,I)/T,
     &      X(203,I)/T*F2,X(204,I)/T,
     &      X(205,I)/T   ,X(206,I)/T,
     &      X(207,I)/T
          ELSE
            WRITE(PR,581) PHS(I),
     &      X(199,I)*F   ,X(200,I)*F,
     &      X(201,I)*F   ,X(202,I)*F,
     &      X(203,I)*F*F2,X(204,I)*F,
     &      X(205,I)*F   ,X(206,I)*F,
     &      X(207,I)*F
          END IF
   15   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,203) LABEL,SLDID
        WRITE(PR,116)
        DO 16 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5821) PHS(I),
     &      LOGFN(X(208,I)*F),LOGFN(X(209,I)*F),
     &      LOGFN(X(210,I)*F),LOGFN(X(211,I)*F),
     &      LOGFN(X(212,I)*F),LOGFN(X(213,I)*F),
     &      LOGFN(X(214,I)*F),LOGFN(X(215,I)*F),
     &      LOGFN(X(216,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,5821) PHS(I),
     &      X(208,I)/T,X(209,I)/T,
     &      X(210,I)/T,X(211,I)/T,
     &      X(212,I)/T,X(213,I)/T,
     &      X(214,I)/T,X(215,I)/T,
     &      X(216,I)/T
          ELSE
            WRITE(PR,582) PHS(I),
     &      X(208,I)*F,X(209,I)*F,
     &      X(210,I)*F,X(211,I)*F,
     &      X(212,I)*F,X(213,I)*F,
     &      X(214,I)*F,X(215,I)*F,
     &      X(216,I)*F
          END IF
   16   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,203) LABEL,SLDID
        WRITE(PR,117)
        DO 17 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6021) PHS(I),
     &      LOGFN(X(217,I)*F),LOGFN(X(218,I)*F),
     &      LOGFN(X(219,I)*F),LOGFN(X(220,I)*F),
     &      LOGFN(X(221,I)*F),LOGFN(X(224,I)*F),
     &      LOGFN(X(223,I)*F),LOGFN(X(291,I)*F),
     &      LOGFN(X(292,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6021) PHS(I),
     &      X(217,I)/T,X(218,I)/T,
     &      X(219,I)/T,X(220,I)/T,
     &      X(221,I)/T,X(224,I)/T,
     &      X(223,I)/T,X(291,I)/T,
     &      X(292,I)/T
          ELSE
            WRITE(PR,602) PHS(I),
     &      X(217,I)*F,X(218,I)*F,
     &      X(219,I)*F,X(220,I)*F,
     &      X(221,I)*F,X(224,I)*F,
     &      X(223,I)*F,X(291,I)*F,
     &      X(292,I)*F
          END IF
   17   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,203) LABEL,SLDID
        WRITE(PR,1171)
        DO I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6024) PHS(I),
     &      LOGFN(X(293,I)*F),LOGFN(TDEDTA(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6025) PHS(I),
     &      X(293,I)/T,TDEDTA(I)/T
          ELSE
            WRITE(PR,6024) PHS(I),
     &      X(293,I)*F,TDEDTA(I)*F
          END IF
        END DO
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=400)
        WRITE(7,202) LABEL
        DO 72 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1)=LOGFN(X(189,I)*F)
            Y(2)=LOGFN(X(190,I)*F)
            Y(3)=LOGFN(X(191,I)*F)
            Y(4)=LOGFN(X(192,I)*F)
            Y(5)=LOGFN(X(193,I)*F)
            Y(6)=LOGFN(X(194,I)*F)
            Y(7)=LOGFN(X(195,I)*F)
            Y(8)=LOGFN(X(196,I)*F)
            Y(9)=LOGFN(X(197,I)*F)
            Y(10)=LOGFN(X(198,I)*F)
            Y(11)=LOGFN(X(199,I)*F)
            Y(12)=LOGFN(X(200,I)*F)
            Y(13)=LOGFN(X(201,I)*F)
            Y(14)=LOGFN(X(202,I)*F)
            Y(15)=LOGFN(X(203,I)*F*F2)
            Y(16)=LOGFN(X(204,I)*F)
            Y(17)=LOGFN(X(205,I)*F)
            Y(18)=LOGFN(X(206,I)*F)
            Y(19)=LOGFN(X(207,I)*F)
            Y(20)=LOGFN(X(208,I)*F)
            Y(21)=LOGFN(X(209,I)*F)
            Y(22)=LOGFN(X(210,I)*F)
            Y(23)=LOGFN(X(211,I)*F)
            Y(24)=LOGFN(X(212,I)*F)
            Y(25)=LOGFN(X(213,I)*F)
            Y(26)=LOGFN(X(214,I)*F)
            Y(27)=LOGFN(X(215,I)*F)
            Y(28)=LOGFN(X(216,I)*F)
            Y(29)=LOGFN(X(217,I)*F)
            Y(30)=LOGFN(X(218,I)*F)
            Y(31)=LOGFN(X(219,I)*F)
            Y(32)=LOGFN(X(220,I)*F)
            Y(33)=LOGFN(X(221,I)*F)
            Y(34)=LOGFN(X(224,I)*F)
            Y(35)=LOGFN(X(223,I)*F)
            Y(36)=LOGFN(X(291,I)*F)
            Y(37)=LOGFN(X(292,I)*F)
            Y(38)=LOGFN(X(292,I)*F)
            Y(39)=LOGFN(TDEDTA(I)*F)
            WRITE(7,6022) PHS(I),(Y(J),J=1,39)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6023) PHS(I),
     &      X(189,I)/T,X(190,I)/T,X(191,I)/T,X(192,I)/T,
     &      X(193,I)/T,X(194,I)/T,X(195,I)/T,X(196,I)/T,
     &      X(197,I)/T,X(198,I)/T,X(199,I)/T,X(200,I)/T,
     &      X(201,I)/T,X(202,I)/T,X(203,I)/T*F2,X(204,I)/T,
     &      X(205,I)/T,X(206,I)/T,X(207,I)/T,
     &      X(208,I)/T,X(209,I)/T,X(210,I)/T,X(211,I)/T,
     &      X(212,I)/T,X(213,I)/T,X(214,I)/T,X(215,I)/T,
     &      X(216,I)/T,X(217,I)/T,X(218,I)/T,
     &      X(219,I)/T,X(220,I)/T,X(221,I)/T,X(224,I)/T,
     &      X(223,I)/T,X(291,I)/T,X(292,I)/T,X(293,I)/T,
     &      TDEDTA(I)/T
          ELSE
            WRITE(7,6022) PHS(I),
     &      X(189,I)*F,X(190,I)*F,X(191,I)*F,X(192,I)*F,
     &      X(193,I)*F,X(194,I)*F,X(195,I)*F,X(196,I)*F,
     &      X(197,I)*F,X(198,I)*F,X(199,I)*F,X(200,I)*F,
     &      X(201,I)*F,X(202,I)*F,X(203,I)*F*F2,X(204,I)*F,
     &      X(205,I)*F,X(206,I)*F,X(207,I)*F,
     &      X(208,I)*F,X(209,I)*F,X(210,I)*F,X(211,I)*F,
     &      X(212,I)*F,X(213,I)*F,X(214,I)*F,X(215,I)*F,
     &      X(216,I)*F,X(217,I)*F,X(218,I)*F,
     &      X(219,I)*F,X(220,I)*F,X(221,I)*F,X(224,I)*F,
     &      X(223,I)*F,X(291,I)*F,X(292,I)*F,X(293,I)*F,
     &      TDEDTA(I)*F
          END IF
   72   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 202  FORMAT(     '       EDTA SYSTEM IN ',A38)
 203  FORMAT(/T44,'       EDTA SYSTEM IN ',A38//T50,'SOLID = ',A27)
 581  FORMAT(/,F5.2,1P3D13.4,D15.4,D17.4,D16.4,3D13.4)
 5811 FORMAT(/,F5.2,  3F13.4,F15.4,F17.4,F16.4,3F13.4)
 582  FORMAT(/,F5.2,1P7D13.4,2D15.4)
 5821 FORMAT(/,F5.2,  7F13.4,2F15.4)
 602  FORMAT(/,F5.2,1P6D13.4,4D12.4)
 6021 FORMAT(/,F5.2,  6F13.4,4F12.4)
 6022 FORMAT(F5.2,39F10.4)
 6023 FORMAT(F5.2,39F6.1)
 6024 FORMAT(/,F5.2,1P6D13.4,4D12.4)
 6025 FORMAT(/,F5.2,  6F14.4,4F12.4)
 114  FORMAT(/'  pH     EDTA(-4)     HEDTA(-3)  ',
     & ' H2EDTA(-2)    H3EDTA(-)    H4EDTA(0)   NaEDTA(-3) ',
     & ' MgEDTA(-2)  MgHEDTA(-)  CaEDTA(-2)  CaHEDTA(-)')
```

```
Line#  Source,Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  256   115  FORMAT(/'  pH     FeEDTA(-)   FeHEDTA(0) ',
  257       & ' FeOHEDTA(-2)  Fe(OH)2EDTA(-3)  (FeOHEDTA)2(-4)  ',
  258       & ' HgEDTA(-2)   HgHEDTA(-)  HgOHEDTA(-3)  CdEDTA(-2)')
  259   116  FORMAT(/'  pH     CdHEDTA(-)   CuEDTA(-2)  ',
  260       & ' CuHEDTA(-)  CuOHEDTA(-3)  FeEDTA(-2)   FeHEDTA(-) ',
  261       & ' FeOHEDTA(-3)  Fe(OH)2EDTA(-4)  MnEDTA(-2)')
  262   117  FORMAT(/'  pH     MnHEDTA(-)   ZnEDTA(-2)  ',
  263       & ' ZnHEDTA(-)  ZnOHEDTA(-3)   KEDTA(-3)   CoEDTA(-2) ',
  264       & ' CoHEDTA(-)   AlEDTA(-)  AlHEDTA(0)')
  265   1171 FORMAT(/'  pH     AlOHEDTA(-2)     T.D.EDTA')
  266   C
  267          RETURN
  268          END


WEDTA  Local Symbols

Name                          Class   Type              Size   Offset

AFILE . . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . . param                             000e
TDEDTA. . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local   INTEGER*4            4    0002
J . . . . . . . . . . . . . . local   INTEGER*4            4    0006
T . . . . . . . . . . . . . . local   REAL*8               8    000a
ISTART. . . . . . . . . . . . local   INTEGER*4            4    0012
Y . . . . . . . . . . . . . . local   REAL*8             312    0016
LABEL . . . . . . . . . . . . local   CHAR*38             38    014e
IEND. . . . . . . . . . . . . local   INTEGER*4            4    0174
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    0178
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020

  269   C
  270          SUBROUTINE WNTA(TDNTA,T1,UNIT,AFILE)
  271   C      THIS BLOCK WRITES THE NTA SUBSYSTEM AQUEOUS SPECIES REPORT.
  272   C
  273          REAL*4 PHS(41),TDNTA(41)
  274          REAL*8 X(319,41),Y(36)
  275          COMMON /OUT/X,PHS
  276          INTEGER I,J,ISTART,IEND,SKIP,UNIT
  277          REAL*8 F,F2,F3,T,T1
  278          INTEGER NUMPH,NPL,PR
```

```
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CARBON               '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CARBON          '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CARBON     '
        WRITE(PR,205) LABEL,SLDID
        WRITE(PR,118)
        DO 18 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5831) PHS(I),
     &      LOGFN(X(236,I)*F),LOGFN(X(237,I)*F),
     &      LOGFN(X(238,I)*F),LOGFN(X(239,I)*F),
     &      LOGFN(X(240,I)*F),LOGFN(X(241,I)*F),
     &      LOGFN(X(242,I)*F),LOGFN(X(243,I)*F*F2),
     &      LOGFN(X(244,I)*F),LOGFN(X(245,I)*F*F2)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,5831) PHS(I),
     &      X(236,I)/T,X(237,I)/T,
     &      X(238,I)/T,X(239,I)/T,
     &      X(240,I)/T,X(241,I)/T,
     &      X(242,I)/T,X(243,I)/T*F2,
     &      X(244,I)/T,X(245,I)/T*F2
          ELSE
            WRITE(PR,583) PHS(I),
     &      X(236,I)*F,X(237,I)*F,
     &      X(238,I)*F,X(239,I)*F,
     &      X(240,I)*F,X(241,I)*F,
     &      X(242,I)*F,X(243,I)*F*F2,
     &      X(244,I)*F,X(245,I)*F*F2
          END IF
   18   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,205) LABEL,SLDID
        WRITE(PR,119)
        DO 19 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5841) PHS(I),
     &      LOGFN(X(246,I)*F)   ,LOGFN(X(247,I)*F),
     &      LOGFN(X(248,I)*F)   ,LOGFN(X(249,I)*F),
     &      LOGFN(X(250,I)*F*F2),LOGFN(X(251,I)*F),
     &      LOGFN(X(252,I)*F)   ,LOGFN(X(253,I)*F*F2),
     &      LOGFN(X(254,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,5841) PHS(I),
     &      X(246,I)/T   ,X(247,I)/T,
     &      X(248,I)/T   ,X(249,I)/T,
     &      X(250,I)/T*F2,X(251,I)/T,
     &      X(252,I)/T   ,X(253,I)/T*F2,
     &      X(254,I)/T
          ELSE
            WRITE(PR,584) PHS(I),
     &      X(246,I)*F   ,X(247,I)*F,
     &      X(248,I)*F   ,X(249,I)*F,
     &      X(250,I)*F*F2,X(251,I)*F,
     &      X(252,I)*F   ,X(253,I)*F*F2,
     &      X(254,I)*F
          END IF
   19   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,205) LABEL,SLDID
        WRITE(PR,120)
        DO 20 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6041) PHS(I),
     &      LOGFN(X(255,I)*F),LOGFN(X(256,I)*F*F2),
     &      LOGFN(X(257,I)*F),LOGFN(X(258,I)*F),
     &      LOGFN(X(259,I)*F),LOGFN(X(260,I)*F*F2),
     &      LOGFN(X(261,I)*F),LOGFN(X(262,I)*F*F2),
     &      LOGFN(X(263,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6041) PHS(I),
     &      X(255,I)/T,X(256,I)/T*F2,
     &      X(257,I)/T,X(258,I)/T,
     &      X(259,I)/T,X(260,I)/T*F2,
     &      X(261,I)/T,X(262,I)/T*F2,
     &      X(263,I)/T
          ELSE
            WRITE(PR,604) PHS(I),
     &      X(255,I)*F,X(256,I)*F*F2,
     &      X(257,I)*F,X(258,I)*F,
     &      X(259,I)*F,X(260,I)*F*F2,
     &      X(261,I)*F,X(262,I)*F*F2,
     &      X(263,I)*F
```

```
          END IF
   20   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,205) LABEL,SLDID
        WRITE(PR,121)
        DO 21 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6041) PHS(I),
     &      LOGFN(X(264,I)*F),LOGFN(X(265,I)*F*F2),
     &      LOGFN(X(266,I)*F),LOGFN(X(294,I)*F),
     &      LOGFN(X(295,I)*F),LOGFN(X(296,I)*F),
     &      LOGFN(X(297,I)*F),LOGFN(TDNTA(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6041) PHS(I),
     &      X(264,I)/T,X(265,I)/T*F2,
     &      X(266,I)/T,X(294,I)/T,
     &      X(295,I)/T,X(296,I)/T,
     &      X(297,I)/T,TDNTA(I)/T
          ELSE
            WRITE(PR,604) PHS(I),
     &      X(264,I)*F,X(265,I)*F*F2,
     &      X(266,I)*F,X(294,I)*F,
     &      X(295,I)*F,X(296,I)*F,
     &      X(297,I)*F,TDNTA(I)*F
          END IF
   21   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &                 RECL=370)
        WRITE(7,204) LABEL
        DO 73 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1)=LOGFN(X(236,I)*F)
            Y(2)=LOGFN(X(237,I)*F)
            Y(3)=LOGFN(X(238,I)*F)
            Y(4)=LOGFN(X(239,I)*F)
            Y(5)=LOGFN(X(240,I)*F)
            Y(6)=LOGFN(X(241,I)*F)
            Y(7)=LOGFN(X(242,I)*F)
            Y(8)=LOGFN(X(243,I)*F*F2)
            Y(9)=LOGFN(X(244,I)*F)
            Y(10)=LOGFN(X(245,I)*F*F2)
            Y(11)=LOGFN(X(246,I)*F)
            Y(12)=LOGFN(X(247,I)*F)
            Y(13)=LOGFN(X(248,I)*F)
            Y(14)=LOGFN(X(249,I)*F)
            Y(15)=LOGFN(X(250,I)*F*F2)
            Y(16)=LOGFN(X(251,I)*F)
            Y(17)=LOGFN(X(252,I)*F)
            Y(18)=LOGFN(X(253,I)*F*F2)
            Y(19)=LOGFN(X(254,I)*F)
            Y(20)=LOGFN(X(255,I)*F)
            Y(21)=LOGFN(X(256,I)*F*F2)
            Y(22)=LOGFN(X(257,I)*F)
            Y(23)=LOGFN(X(258,I)*F)
            Y(24)=LOGFN(X(259,I)*F)
            Y(25)=LOGFN(X(260,I)*F*F2)
            Y(26)=LOGFN(X(261,I)*F)
            Y(27)=LOGFN(X(262,I)*F*F2)
            Y(28)=LOGFN(X(263,I)*F)
            Y(29)=LOGFN(X(264,I)*F)
            Y(30)=LOGFN(X(265,I)*F*F2)
            Y(31)=LOGFN(X(266,I)*F)
            Y(32)=LOGFN(X(294,I)*F)
            Y(33)=LOGFN(X(295,I)*F)
            Y(34)=LOGFN(X(296,I)*F)
            Y(35)=LOGFN(X(297,I)*F)
            Y(36)=LOGFN(TDNTA(I)*F)
            WRITE(7,6042) PHS(I),(Y(J),J=1,36)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6043) PHS(I),
     &      X(236,I)/T   ,X(237,I)/T,   X(238,I)/T   ,X(239,I)/T,
     &      X(240,I)/T   ,X(241,I)/T,   X(242,I)/T   ,X(243,I)/T*F2,
     &      X(244,I)/T   ,X(245,I)/T*F2,X(246,I)/T   ,X(247,I)/T,
     &      X(248,I)/T   ,X(249,I)/T,   X(250,I)/T*F2,X(251,I)/T,
     &      X(252,I)/T   ,X(253,I)/T*F2,X(254,I)/T,
     &      X(255,I)/T   ,X(256,I)/T*F2,X(257,I)/T   ,X(258,I)/T,
     &      X(259,I)/T   ,X(260,I)/T*F2,X(261,I)/T   ,X(262,I)/T*F2,
     &      X(263,I)/T,   X(264,I)/T   ,X(265,I)/T*F2,
     &      X(266,I)/T   ,X(294,I)/T   ,X(295,I)/T   ,X(296,I)/T,
     &      X(297,I)/T   ,TDNTA(I)/T
          ELSE
            WRITE(7,6042) PHS(I),
     &      X(236,I)*F,   X(237,I)*F,   X(238,I)*F,   X(239,I)*F,
     &      X(240,I)*F,   X(241,I)*F,   X(242,I)*F,   X(243,I)*F*F2,
     &      X(244,I)*F,   X(245,I)*F*F2,X(246,I)*F,   X(247,I)*F,
     &      X(248,I)*F,   X(249,I)*F,   X(250,I)*F*F2,X(251,I)*F,
     &      X(252,I)*F,   X(253,I)*F*F2,X(254,I)*F,
     &      X(255,I)*F,   X(256,I)*F*F2,X(257,I)*F,   X(258,I)*F,
     &      X(259,I)*F,   X(260,I)*F*F2,X(261,I)*F,   X(262,I)*F*F2,
     &      X(263,I)*F,   X(264,I)*F,   X(265,I)*F*F2,
     &      X(266,I)*F,   X(294,I)*F,   X(295,I)*F,   X(296,I)*F,
     &      X(297,I)*F,   TDNTA(I)*F
          END IF
   73   CONTINUE
        CLOSE (UNIT=7)
        END IF
C
 204   FORMAT(      '         NTA SYSTEM IN ',A38)
 205   FORMAT(/T44,'         NTA SYSTEM IN ',A38//T50,'SOLID = ',A27)
 583   FORMAT(/,F5.2,1P7D12.4,3D14.4)
 5831  FORMAT(/,F5.2,  7F12.4,3F14.4)
 584   FORMAT(/,F5.2,1PD13.4,D15.4,D14.4,6D13.4)
 5841  FORMAT(/,F5.2,  F13.4,F15.4,F14.4,6F13.4)
 604   FORMAT(/,F5.2,1P5D13.4,D12.4,D14.4,2D13.4)
 6041  FORMAT(/,F5.2,  5F13.4,F12.4,D14.4,2D13.4)
 6042  FORMAT(F5.2,36F10.4)
 6043  FORMAT(F5.2,36F6.1)
 118   FORMAT(/'  pH      NTA(-3)    HNTA(-2)   ',
     & ' H2NTA(-)    H3NTA(0)    NaNTA(-2)   MgNTA(-)    ',
     & ' CaNTA(-)    Ca(NTA)2(-4)    FeNTA(0)    Fe(NTA)2(-3)')
 119   FORMAT(/'  pH    FeOHNTA(-)   Fe(OH)2NTA(-2)   ',
     & ' HgNTA(-)     CdNTA(-)   Cd(NTA)2(-4)  CdOHNTA(-2)   ',
     & ' CuNTA(-)   Cu(NTA)2(-4)  CuOHNTA(-2)')
 120   FORMAT(/'  pH      FeNTA(-)   Fe(NTA)2(-4)   ',
     & ' FeHNTA(0)   FeOHNTA(-2)   MnNTA(-)   Mn(NTA)2(-4)   ',
     & ' ZnNTA(-)   Zn(NTA)2(-4)  ZnOHNTA(-2)')
 121   FORMAT(/'  pH      CoNTA(-)   Co(NTA)2(-4) ',
     & ' CoOHNTA(-2)    AlNTA(0)    AlHNTA(+)  AlOHNTA(-)',
     & '  Al(OH)2NTA(-2)  T.D.NTA')
C
        RETURN
        END
```

```
WNTA  Local Symbols

Name                         Class   Type              Size   Offset

AFILE . . . . . . . . . . . param                              0006
UNIT. . . . . . . . . . . . param                              000a
T1. . . . . . . . . . . . . param                              000e
TDNTA . . . . . . . . . . . param                              0012
I . . . . . . . . . . . . . local   INTEGER*4            4     017c
J . . . . . . . . . . . . . local   INTEGER*4            4     0180
T . . . . . . . . . . . . . local   REAL*8               8     0184
ISTART. . . . . . . . . . . local   INTEGER*4            4     018c
Y . . . . . . . . . . . . . local   REAL*8             288     0190
LABEL . . . . . . . . . . . local   CHAR*38             38     02b0
IEND. . . . . . . . . . . . local   INTEGER*4            4     02d6
SKIP. . . . . . . . . . . . local   INTEGER*4            4     02da
PHS . . . . . . . . . . . . OUT     REAL*4             164     98b8
X . . . . . . . . . . . . . OUT     REAL*8          104632     0000
F . . . . . . . . . . . . . W       REAL*8               8     0000
F2. . . . . . . . . . . . . W       REAL*8               8     0008
F3. . . . . . . . . . . . . W       REAL*8               8     0010
NUMPH . . . . . . . . . . . W       INTEGER*4            4     0018
```

```
WNTA  Local Symbols

Name                        Class   Type               Size   Offset

NPL . . . . . . . . . . . . . W       INTEGER*4             4     001c
PR. . . . . . . . . . . . . . W       INTEGER*4             4     003c
SLDID . . . . . . . . . . . . W       CHAR*27              27     0020
```

```
C
      SUBROUTINE WC(TDIC,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE CARBON SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDIC(41)
      REAL*8 X(319,41),Y(13)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CARBON               '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CARBON          '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CARBON     '
        WRITE(PR,207) LABEL,SLDID
        WRITE(PR,122)
        DO 22 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6281) PHS(I),
     &      LOGFN(X(  1,I)*F),LOGFN(X(  2,I)*F),
     &      LOGFN(X(  4,I)*F),LOGFN(X(  9,I)*F),
     &      LOGFN(X( 12,I)*F),LOGFN(X(  7,I)*F),
     &      LOGFN(X( 11,I)*F),LOGFN(X( 19,I)*F),
     &      LOGFN(X( 27,I)*F),LOGFN(X( 20,I)*F*F2)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6281) PHS(I),
     &      X(  1,I)/T,X(  2,I)/T,
     &      X(  4,I)/T,X(  9,I)/T,
     &      X( 12,I)/T,X(  7,I)/T,
     &      X( 11,I)/T,X( 19,I)/T,
     &      X( 27,I)/T,X( 20,I)/T*F2
          ELSE
            WRITE(PR,628) PHS(I),
     &      X(  1,I)*F,X(  2,I)*F,
     &      X(  4,I)*F,X(  9,I)*F,
     &      X( 12,I)*F,X(  7,I)*F,
     &      X( 11,I)*F,X( 19,I)*F,
     &      X( 27,I)*F,X( 20,I)*F*F2
          END IF
   22   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,207) LABEL,SLDID
        WRITE(PR,123)
        DO 23 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6281) PHS(I),
     &      LOGFN(X(113,I)*F),LOGFN(X(299,I)*F),
     &      LOGFN(TDIC(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6281) PHS(I),
     &      X(113,I)/T,X(299,I)/T,
     &      TDIC(I)/T
          ELSE
            WRITE(PR,628) PHS(I),
     &      X(113,I)*F,X(299,I)*F,
     &      TDIC(I)*F
          END IF
   23   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &              RECL=130)
        WRITE(7,206) LABEL
        DO 74 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(  1,I)*F)
            Y(2 )=LOGFN(X(  2,I)*F)
            Y(3 )=LOGFN(X(  4,I)*F)
            Y(4 )=LOGFN(X(  9,I)*F)
            Y(5 )=LOGFN(X( 12,I)*F)
            Y(6 )=LOGFN(X(  7,I)*F)
            Y(7 )=LOGFN(X( 11,I)*F)
            Y(8 )=LOGFN(X( 19,I)*F)
```

```
  603                 Y(9 )=LOGFN(X( 27,I)*F)
  604                 Y(10)=LOGFN(X( 20,I)*F*F2)
  605                 Y(11)=LOGFN(X(113,I)*F)
  606                 Y(12)=LOGFN(X(299,I)*F)
  607                 Y(13)=LOGFN(TDIC(I)*F)
  608                 WRITE(7,6282) PHS(I),(Y(J),J=1,13)
  609               ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
  610                 WRITE(7,6283) PHS(I),
  611         &       X(  1,I)/T,X(  2,I)/T,
  612         &       X(  4,I)/T,X(  9,I)/T,
  613         &       X( 12,I)/T,X(  7,I)/T,
  614         &       X( 11,I)/T,X( 19,I)/T,
  615         &       X( 27,I)/T,X( 20,I)/T*F2,
  616         &       X(113,I)/T,X(299,I)/T,
  617         &       TDIC(I)/T
  618               ELSE
  619                 WRITE(7,6282) PHS(I),
  620         &       X(  1,I)*F,X(  2,I)*F,
  621         &       X(  4,I)*F,X(  9,I)*F,
  622         &       X( 12,I)*F,X(  7,I)*F,
  623         &       X( 11,I)*F,X( 19,I)*F,
  624         &       X( 27,I)*F,X( 20,I)*F*F2,
  625         &       X(113,I)*F,X(299,I)*F,
  626         &       TDIC(I)*F
  627               END IF
  628      74     CONTINUE
  629             CLOSE (UNIT=7)
  630           END IF
  631   C
  632    206  FORMAT(      'INORGANIC CARBON SYSTEM IN ',A38)
  633    207  FORMAT(/T44,'INORGANIC CARBON SYSTEM IN ',A38//T50,'SOLID = ',A27
  634    628  FORMAT(/,F5.2,1P7D12.4,D13.4,2D14.4)
  635    6281 FORMAT(/,F5.2,  7F12.4,F13.4,2F14.4)
  636    6282 FORMAT(F5.2,13F10.4)
  637    6283 FORMAT(F5.2,13F6.1)
  638    122  FORMAT(/'  pH      CO3(-2)     HCO3(-)   ',
  639         & ' H2CO3(0)     MgCO3(0)    MgHCO3(+)   CaCO3(0)    ',
  640         & ' CaHCO3(+)    CuCO3(0)       CuHCO3(+)    Cu(CO3)2(-2)')
  641    123  FORMAT(/'  pH    MnHCO3(+)     AlCO3(+)     T.D.IC')
  642   C
  643           RETURN
  644           END


WC  Local Symbols

Name                           Class   Type              Size   Offset

AFILE . . . . . . . . . . . . param                              0006
UNIT. . . . . . . . . . . . . param                              000a
T1. . . . . . . . . . . . . . param                              000e
```

```
WC  Local Symbols

Name                      Class   Type              Size   Offset

TDIC. . . . . . . . . . . param                            0012
I . . . . . . . . . . . . local   INTEGER*4            4   02de
J . . . . . . . . . . . . local   INTEGER*4            4   02e2
T . . . . . . . . . . . . local   REAL*8               8   02e6
ISTART. . . . . . . . . . local   INTEGER*4            4   02ee
Y . . . . . . . . . . . . local   REAL*8             104   02f2
LABEL . . . . . . . . . . local   CHAR*38             38   035a
IEND. . . . . . . . . . . local   INTEGER*4            4   0380
SKIP. . . . . . . . . . . local   INTEGER*4            4   0384
PHS . . . . . . . . . . . OUT     REAL*4             164   98b8
X . . . . . . . . . . . . OUT     REAL*8          104632   0000
F . . . . . . . . . . . . W       REAL*8               8   0000
F2. . . . . . . . . . . . W       REAL*8               8   0008
F3. . . . . . . . . . . . W       REAL*8               8   0010
NUMPH . . . . . . . . . . W       INTEGER*4            4   0018
NPL . . . . . . . . . . . W       INTEGER*4            4   001c
PR. . . . . . . . . . . . W       INTEGER*4            4   003c
SLDID . . . . . . . . . . W       CHAR*27             27   0020

Global Symbols

Name                      Class   Type              Size   Offset

LOGFN . . . . . . . . . . extern  REAL*8             ***      ***
OUT . . . . . . . . . . . common  ***             104796     0000
W . . . . . . . . . . . . common  ***                 64     0000
WC. . . . . . . . . . . . FSUBRT  ***                ***     a8d5
WEDTA . . . . . . . . . . FSUBRT  ***                ***     0000
WFOOT . . . . . . . . . . extern  ***                ***      ***
WHEAD . . . . . . . . . . extern  ***                ***      ***
WNTA. . . . . . . . . . . FSUBRT  ***                ***     5742

Code size = c783 (51075)
Data size = 07ab (1963)
Bss size  = 0388 (904)

No errors detected
```

```
      SUBROUTINE WCL(TDCL,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE CLORIDE SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDCL(41)
      REAL*8 X(319,41),Y(19)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CHLORINE             '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CHLORINE        '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CHLORINE   '
        WRITE(PR,209) LABEL,SLDID
        WRITE(PR,124)
        DO 24 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6301) PHS(I),
     &      LOGFN(X( 16,I)*F)   ,LOGFN(X(160,I)*F),
     &      LOGFN(X(161,I)*F*F2),LOGFN(X(162,I)*F*F3),
     &      LOGFN(X( 89,I)*F)   ,LOGFN(X( 90,I)*F),
     &      LOGFN(X( 91,I)*F*F2),LOGFN(X( 92,I)*F*F3),
     &      LOGFN(X( 47,I)*F)   ,LOGFN(X( 48,I)*F*F2)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6301) PHS(I),
     &      X( 16,I)/T   ,X(160,I)/T,
     &      X(161,I)/T*F2,X(162,I)/T*F3,
     &      X( 89,I)/T   ,X( 90,I)/T,
     &      X( 91,I)/T*F2,X( 92,I)/T*F3,
     &      X( 47,I)/T   ,X( 48,I)/T*F2
          ELSE
            WRITE(PR,630) PHS(I),
     &      X( 16,I)*F   ,X(160,I)*F,
     &      X(161,I)*F*F2,X(162,I)*F*F3,
     &      X( 89,I)*F   ,X( 90,I)*F,
     &      X( 91,I)*F*F2,X( 92,I)*F*F3,
     &      X( 47,I)*F   ,X( 48,I)*F*F2
```

```
          END IF
   24   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,209) LABEL,SLDID
        WRITE(PR,125)
        DO 25 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6301) PHS(I),
     &      LOGFN(X( 49,I)*F*F3),LOGFN(X( 28,I)*F),
     &      LOGFN(X( 29,I)*F*F2),LOGFN(X(117,I)*F),
     &      LOGFN(X(131,I)*F)   ,LOGFN(X(132,I)*F*F2),
     &      LOGFN(X(133,I)*F*F3),LOGFN(X(148,I)*F),
     &      LOGFN(TDCL(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6301) PHS(I),
     &      X( 49,I)/T*F3,X( 28,I)/T,
     &      X( 29,I)/T*F2,X(117,I)/T,
     &      X(131,I)/T   ,X(132,I)/T*F2,
     &      X(133,I)/T*F3,X(148,I)/T,
     &      TDCL(I)/T
          ELSE
            WRITE(PR,630) PHS(I),
     &      X( 49,I)*F*F3,X( 28,I)*F,
     &      X( 29,I)*F*F2,X(117,I)*F,
     &      X(131,I)*F   ,X(132,I)*F*F2,
     &      X(133,I)*F*F3,X(148,I)*F,
     &      TDCL(I)*F
          END IF
   25   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=200)
        WRITE(7,208) LABEL
        DO 75 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X( 16,I)*F)
            Y(2 )=LOGFN(X(160,I)*F)
            Y(3 )=LOGFN(X(161,I)*F*F2)
            Y(4 )=LOGFN(X(162,I)*F*F3)
            Y(5 )=LOGFN(X( 89,I)*F)
            Y(6 )=LOGFN(X( 90,I)*F)
            Y(7 )=LOGFN(X( 91,I)*F*F2)
            Y(8 )=LOGFN(X( 92,I)*F*F3)
            Y(9 )=LOGFN(X( 47,I)*F)
            Y(10)=LOGFN(X( 48,I)*F*F2)
            Y(11)=LOGFN(X( 49,I)*F*F3)
            Y(12)=LOGFN(X( 28,I)*F)
            Y(13)=LOGFN(X( 29,I)*F*F2)
            Y(14)=LOGFN(X(117,I)*F)
            Y(15)=LOGFN(X(131,I)*F)
            Y(16)=LOGFN(X(132,I)*F*F2)
            Y(17)=LOGFN(X(133,I)*F*F3)
            Y(18)=LOGFN(X(148,I)*F)
            Y(19)=LOGFN(TDCL(I)*F)
            WRITE(7,6302) PHS(I),(Y(J),J=1,19)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6303) PHS(I),
     &      X( 16,I)/T   ,X(160,I)/T,
     &      X(161,I)/T*F2,X(162,I)/T*F3,
     &      X( 89,I)/T   ,X( 90,I)/T,
     &      X( 91,I)/T*F2,X( 92,I)/T*F3,
     &      X( 47,I)/T   ,X( 48,I)/T*F2,
     &      X( 49,I)/T*F3,X( 28,I)/T,
     &      X( 29,I)/T*F2,X(117,I)/T,
     &      X(131,I)/T   ,X(132,I)/T*F2,
     &      X(133,I)/T*F3,X(148,I)/T,
     &      TDCL(I)/T
          ELSE
            WRITE(7,6302) PHS(I),
     &      X( 16,I)*F   ,X(160,I)*F,
     &      X(161,I)*F*F2,X(162,I)*F*F3,
     &      X( 89,I)*F   ,X( 90,I)*F,
     &      X( 91,I)*F*F2,X( 92,I)*F*F3,
     &      X( 47,I)*F   ,X( 48,I)*F*F2,
     &      X( 49,I)*F*F3,X( 28,I)*F,
     &      X( 29,I)*F*F2,X(117,I)*F,
     &      X(131,I)*F   ,X(132,I)*F*F2,
     &      X(133,I)*F*F3,X(148,I)*F,
     &      TDCL(I)*F
          END IF
   75   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 208  FORMAT(     '     CHLORINE SYSTEM IN ',A38)
 209  FORMAT(/T44,'     CHLORINE SYSTEM IN ',A38//T50,'SOLID = ',A27)
 630  FORMAT(/,F5.2,1P6D13.4,4D12.4)
 6301 FORMAT(/,F5.2,  6F13.4,4F12.4)
 6302 FORMAT(F5.2,19F10.4)
 6303 FORMAT(F5.2,19F6.1)
 124  FORMAT(/'  pH       Cl(-)      FeCl(+2)     ',
     & ' FeCl2(+)     FeCl3(0)    Hg(OH)Cl(0)    HgCl(+)     ',
     & ' HgCl2(0)    HgCl3(-)     CdCl(+)    CdCl2(0)')
 125  FORMAT(/'  pH     CdCl3(-)      CuCl(+)     ',
     & ' CuCl2(0)      MnCl(+)      ZnCl(+)     ZnCl2(0)     ',
   154        & ' ZnCl3(-)     KCl(0)      T.D.Cl')
   155  C
   156         RETURN
   157         END

WCL  Local Symbols

Name                         Class   Type              Size   Offset

AFILE . . . . . . . . . . . . param                            0006
UNIT. . . . . . . . . . . . . param                            000a
T1. . . . . . . . . . . . . . param                            000e
TDCL. . . . . . . . . . . . . param                            0012
I . . . . . . . . . . . . . . local  INTEGER*4            4    0002
J . . . . . . . . . . . . . . local  INTEGER*4            4    0006
T . . . . . . . . . . . . . . local  REAL*8               8    000a
ISTART. . . . . . . . . . . . local  INTEGER*4            4    0012
Y . . . . . . . . . . . . . . local  REAL*8             152    0016
LABEL . . . . . . . . . . . . local  CHAR*38             38    00ae
IEND. . . . . . . . . . . . . local  INTEGER*4            4    00d4
SKIP. . . . . . . . . . . . . local  INTEGER*4            4    00d8
PHS . . . . . . . . . . . . . OUT    REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT    REAL*8          104632    0000
F . . . . . . . . . . . . . . W      REAL*8               8    0000
F2. . . . . . . . . . . . . . W      REAL*8               8    0008
F3. . . . . . . . . . . . . . W      REAL*8               8    0010
NUMPH . . . . . . . . . . . . W      INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W      INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W      INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W      CHAR*27             27    0020

   158  C
   159         SUBROUTINE WOC(TDOC,T1,UNIT,AFILE)
   160  C      THIS BLOCK WRITES THE ORGANIC CARBON SUBSYSTEM
   161  C      AQUEOUS SPECIES REPORT.
   162  C
   163         REAL*4 PHS(41),TDOC(41)
   164         REAL*8 X(319,41),Y(7)
   165         COMMON /OUT/X,PHS
   166         INTEGER I,ISTART,IEND,SKIP,UNIT
   167         REAL*8 F,F2,F3,T,T1
   168         INTEGER NUMPH,NPL,PR
   169         CHARACTER LABEL*38,SLDID*27
   170         CHARACTER AFILE*15
   171         REAL*8 LOGFN
   172         COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
   173  C
   174         ISTART=1
   175         IEND=0
   176         F2=F/6.76D-2
```

```
      F3=F/3.38D-2
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF CARBON              '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF CARBON         '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF CARBON    '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CARBON                 '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CARBON            '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CARBON       '
        WRITE(PR,211) LABEL,SLDID
        WRITE(PR,126)
        DO 26 I=ISTART,IEND
          DUMMY=1D1**((PHS(I)-4.86D0)/1.90D0)
          ACID=(1D0+DUMMY)/(1D0+1D0/DUMMY)*X(71,I)
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6311) PHS(I),
     &      LOGFN(ACID*F2)    ,LOGFN(X( 71,I)*F2),
     &      LOGFN(X(102,I)*F3),LOGFN(X( 63,I)*F3),
     &      LOGFN(X( 30,I)*F3),LOGFN(X(305,I)*F3),
     &      LOGFN(TDOC(I)*F2)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6311) PHS(I),
     &      ACID/T    ,X( 71,I)/T,
     &      X(102,I)/T,X( 63,I)/T,
     &      X( 30,I)/T,X(305,I)/T,
     &      TDOC(I)/T
          ELSE
            WRITE(PR,631) PHS(I),
     &      ACID*F2    ,X( 71,I)*F2,
     &      X(102,I)*F3,X( 63,I)*F3,
     &      X( 30,I)*F3,X(305,I)*F3,
     &      TDOC(I)*F2
          END IF
   26   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=70)
        WRITE(7,210) LABEL
        DO 76 I=1,NUMPH
          DUMMY=1D1**((PHS(I)-4.86D0)/1.90D0)
          ACID=(1D0+DUMMY)/(1D0+1D0/DUMMY)*X(71,I)
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
```

```
Line#  Source Line           Microsoft FORTRAN Optimizing Compiler Version 5.00
  228                 Y(1 )=LOGFN(ACID*F2)
  229                 Y(2 )=LOGFN(X( 71,I)*F2)
  230                 Y(3 )=LOGFN(X(102,I)*F3)
  231                 Y(4 )=LOGFN(X( 63,I)*F3)
  232                 Y(5 )=LOGFN(X( 30,I)*F3)
  233                 Y(6 )=LOGFN(X(305,I)*F3)
  234                 Y(7 )=LOGFN(TDOC(I)*F2)
  235                 WRITE(7,6312) PHS(I),(Y(J),J=1,7)
  236               ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
  237                 WRITE(7,6313) PHS(I),
  238         &       ACID/T    ,X( 71,I)/T,
  239         &       X(102,I)/T,X( 63,I)/T,
  240         &       X( 30,I)/T,X(305,I)/T,
  241         &       TDOC(I)/T
  242               ELSE
  243                 WRITE(7,6312) PHS(I),
  244         &       ACID*F2    ,X( 71,I)*F2,
  245         &       X(102,I)*F3,X( 63,I)*F3,
  246         &       X( 30,I)*F3,X(305,I)*F3,
  247         &       TDOC(I)*F2
  248               END IF
  249        76   CONTINUE
  250             CLOSE (UNIT=7)
  251           END IF
  252  C
  253   210   FORMAT(     'ORGANIC CARBON SYSTEM IN ',A38)
  254   211   FORMAT(/T44,'ORGANIC CARBON SYSTEM IN ',A38//T50,'SOLID = ',A27)
  255   631   FORMAT(/,F5.2,1P7D13.4)
  256   6311 FORMAT(/,F5.2,  7F13.4)
  257   6312 FORMAT(F5.2,7F10.4)
  258   6313 FORMAT(F5.2,7F6.1)
  259   126   FORMAT(/'  pH        A(-)        HA(0)       ',
  260        & ' HgA2(0)      CdA2(0)      CuA2(0)      AlA2(0)      T.D.OC')
  261  C
  262         RETURN
  263         END
```

```
WOC  Local Symbols

Name                         Class   Type              Size   Offset

AFILE . . . . . . . . . . . . param                            0006
UNIT. . . . . . . . . . . . . param                            000a
T1. . . . . . . . . . . . . . param                            000e
TDOC. . . . . . . . . . . . . param                            0012
I . . . . . . . . . . . . . . local   INTEGER*4            4   00dc
J . . . . . . . . . . . . . . local   INTEGER*4            4   00e0
DUMMY . . . . . . . . . . . . local   REAL*4               4   00e4
ACID. . . . . . . . . . . . . local   REAL*4               4   00e8
T . . . . . . . . . . . . . . local   REAL*8               8   00ec
```

```
WOC  Local Symbols

Name                          Class   Type              Size   Offset

ISTART. . . . . . . . . . . . local   INTEGER*4            4    00f4
Y . . . . . . . . . . . . . . local   REAL*8              56    00f8
LABEL . . . . . . . . . . . . local   CHAR*38             38    0130
IEND. . . . . . . . . . . . . local   INTEGER*4            4    0156
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    015a
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020
```

```
C
      SUBROUTINE WSE(TDSE,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE SELENITE SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDSE(41)
      REAL*8 X(319,41),Y(9)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES               '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES          '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES     '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF SELENIUM                 '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF SELENIUM            '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF SELENIUM       '
        WRITE(PR,213) LABEL,SLDID
        WRITE(PR,128)
        DO 28 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5851) PHS(I),
     &      LOGFN(X(104,I)*F)   ,LOGFN(X(105,I)*F),
     &      LOGFN(X(108,I)*F*F2),LOGFN(X(106,I)*F),
     &      LOGFN(X(107,I)*F*F2),LOGFN(X(166,I)*F),
     &      LOGFN(X(109,I)*F*F2),LOGFN(X(110,I)*F*F2),
     &      LOGFN(TDSE(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,5851) PHS(I),
     &      X(104,I)/T   ,X(105,I)/T,
     &      X(108,I)/T*F2,X(106,I)/T,
     &      X(107,I)/T*F2,X(166,I)/T,
     &      X(109,I)/T*F2,X(110,I)/T*F2,
     &      TDSE(I)/T
          ELSE
            WRITE(PR,585) PHS(I),
     &      X(104,I)*F   ,X(105,I)*F,
     &      X(108,I)*F*F2,X(106,I)*F,
     &      X(107,I)*F*F2,X(166,I)*F,
     &      X(109,I)*F*F2,X(110,I)*F*F2,
     &      TDSE(I)*F
          END IF
   28   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=100)
        WRITE(7,212) LABEL
        DO 77 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(104,I)*F)
            Y(2 )=LOGFN(X(105,I)*F)
            Y(3 )=LOGFN(X(108,I)*F*F2)
            Y(4 )=LOGFN(X(106,I)*F)
            Y(5 )=LOGFN(X(107,I)*F*F2)
            Y(6 )=LOGFN(X(166,I)*F)
            Y(7 )=LOGFN(X(109,I)*F*F2)
            Y(8 )=LOGFN(X(110,I)*F*F2)
            Y(9 )=LOGFN(TDSE(I)*F)
            WRITE(7,5852) PHS(I),(Y(J),J=1,9)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,5853) PHS(I),
     &      X(104,I)/T   ,X(105,I)/T,
     &      X(108,I)/T*F2,X(106,I)/T,
     &      X(107,I)/T*F2,X(166,I)/T,
     &      X(109,I)/T*F2,X(110,I)/T*F2,
     &      TDSE(I)/T
```

```
  346              ELSE
  347                WRITE(7,5852) PHS(I),
  348         &      X(104,I)*F   ,X(105,I)*F,
  349         &      X(108,I)*F*F2,X(106,I)*F,
  350         &      X(107,I)*F*F2,X(166,I)*F,
  351         &      X(109,I)*F*F2,X(110,I)*F*F2,
  352         &      TDSE(I)*F
  353              END IF
  354       77   CONTINUE
  355            CLOSE (UNIT=7)
  356          END IF
  357   C
  358    212   FORMAT(     '    SELENITE SYSTEM IN ',A38)
  359    213   FORMAT(/T44,'    SELENITE SYSTEM IN ',A38//T50,'SOLID = ',A27)
  360    585   FORMAT(/,F5.2,1P2D13.4,3D14.4,3D15.4,D14.4)
  361    5851 FORMAT(/,F5.2,  2F13.4,3F14.4,3F15.4,F14.4)
  362    5852 FORMAT(F5.2,9F10.4)
  363    5853 FORMAT(F5.2,9F6.1)
  364    128   FORMAT(/'  pH     SeO3(-2)    HSeO3(-)   ',
  365         & ' H(SeO3)2(-3)    H2SeO3(0)   H2(SeO3)2(-2)   FeHSeO3(+2)  ',
  366         & ' Hg(SeO3)2(-2)  Cd(SeO3)2(-2)   T.D.SeO3')
  367   C
  368          RETURN
  369          END
```

```
WSE  Local Symbols

Name                      Class   Type              Size   Offset

AFILE . . . . . . . . . . param                            0006
UNIT. . . . . . . . . . . param                            000a
T1. . . . . . . . . . . . param                            000e
TDSE. . . . . . . . . . . param                            0012
I . . . . . . . . . . . . local   INTEGER*4            4   015e
J . . . . . . . . . . . . local   INTEGER*4            4   0162
T . . . . . . . . . . . . local   REAL*8               8   0166
ISTART. . . . . . . . . . local   INTEGER*4            4   016e
Y . . . . . . . . . . . . local   REAL*8              72   0172
LABEL . . . . . . . . . . local   CHAR*38             38   01ba
IEND. . . . . . . . . . . local   INTEGER*4            4   01e0
SKIP. . . . . . . . . . . local   INTEGER*4            4   01e4
PHS . . . . . . . . . . . OUT     REAL*4             164   98b8
X . . . . . . . . . . . . OUT     REAL*8          104632   0000
F . . . . . . . . . . . . W       REAL*8               8   0000
F2. . . . . . . . . . . . W       REAL*8               8   0008
F3. . . . . . . . . . . . W       REAL*8               8   0010
NUMPH . . . . . . . . . . W       INTEGER*4            4   0018
NPL . . . . . . . . . . . W       INTEGER*4            4   001c
PR. . . . . . . . . . . . W       INTEGER*4            4   003c
SLDID . . . . . . . . . . W       CHAR*27             27   0020
```

```
C
      SUBROUTINE WSIO4(TDSIO4,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE SILICATE SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDSIO4(41)
      REAL*8 X(319,41),Y(14)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MINO(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF SILICON              '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF SILICON         '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF SILICON    '
        WRITE(PR,215) LABEL,SLDID
        WRITE(PR,129)
        DO 29 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5861) PHS(I),
     &      LOGFN(X(186,I)*F)   ,LOGFN(X(169,I)*F),
     &      LOGFN(X(168,I)*F)   ,LOGFN(X(268,I)*F),
     &      LOGFN(X(269,I)*F)   ,LOGFN(X(170,I)*F*F2),
     &      LOGFN(X(173,I)*F)   ,LOGFN(X(174,I)*F),
     &      LOGFN(X(175,I)*F*F2)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,5861) PHS(I),
     &      X(186,I)/T   ,X(169,I)/T,
     &      X(168,I)/T   ,X(268,I)/T,
     &      X(269,I)/T   ,X(170,I)/T*F2,
     &      X(173,I)/T   ,X(174,I)/T,
     &      X(175,I)/T*F2
          ELSE
            WRITE(PR,586) PHS(I),
     &      X(186,I)*F   ,X(169,I)*F,
     &      X(168,I)*F   ,X(268,I)*F,
     &      X(269,I)*F   ,X(170,I)*F*F2,
     &      X(173,I)*F   ,X(174,I)*F,
     &      X(175,I)*F*F2
          END IF
   29   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,215) LABEL,SLDID
        WRITE(PR,130)
        DO 30 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5871) PHS(I),
     &      LOGFN(X(176,I)*F)   ,LOGFN(X(177,I)*F),
     &      LOGFN(X(178,I)*F*F2),LOGFN(X(167,I)*F),
     &      LOGFN(TDSIO4(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,5871) PHS(I),
     &      X(176,I)/T   ,X(177,I)/T,
     &      X(178,I)/T*F2,X(167,I)/T,
     &      TDSIO4(I)/T
          ELSE
            WRITE(PR,587) PHS(I),
     &      X(176,I)*F   ,X(177,I)*F,
     &      X(178,I)*F*F2,X(167,I)*F,
     &      TDSIO4(I)*F
          END IF
   30   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=150)
        WRITE(7,214) LABEL
        DO 78 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(186,I)*F)
            Y(2 )=LOGFN(X(169,I)*F)
            Y(3 )=LOGFN(X(168,I)*F)
            Y(4 )=LOGFN(X(268,I)*F)
            Y(5 )=LOGFN(X(269,I)*F)
            Y(6 )=LOGFN(X(170,I)*F*F2)
            Y(7 )=LOGFN(X(173,I)*F)
            Y(8 )=LOGFN(X(174,I)*F)
            Y(9 )=LOGFN(X(175,I)*F*F2)
            Y(10)=LOGFN(X(176,I)*F)
            Y(11)=LOGFN(X(177,I)*F)
            Y(12)=LOGFN(X(178,I)*F*F2)
```

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00
  471                 Y(13)=LOGFN(X(167,I)*F)
  472                 Y(14)=LOGFN(TDSIO4(I)*F)
  473                 WRITE(7,5872) PHS(I),(Y(J),J=1,14)
  474               ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
  475                 WRITE(7,5873) PHS(I),
  476         &       X(186,I)/T   ,X(169,I)/T,
  477         &       X(168,I)/T   ,X(268,I)/T,
  478         &       X(269,I)/T   ,X(170,I)/T*F2,
  479         &       X(173,I)/T   ,X(174,I)/T,
  480         &       X(175,I)/T*F2,
  481         &       X(176,I)/T   ,X(177,I)/T,
  482         &       X(178,I)/T*F2,X(167,I)/T,
  483         &       TDSIO4(I)/T
  484               ELSE
  485                 WRITE(7,5872) PHS(I),
  486         &       X(186,I)*F   ,X(169,I)*F,
  487         &       X(168,I)*F   ,X(268,I)*F,
  488         &       X(269,I)*F   ,X(170,I)*F*F2,
  489         &       X(173,I)*F   ,X(174,I)*F,
  490         &       X(175,I)*F*F2,
  491         &       X(176,I)*F   ,X(177,I)*F,
  492         &       X(178,I)*F*F2,X(167,I)*F,
  493         &       TDSIO4(I)*F
  494               END IF
  495      78     CONTINUE
  496             CLOSE (UNIT=7)
  497           END IF
  498     C
  499      214  FORMAT(     '     SILICATE SYSTEM IN ',A38)
  500      215  FORMAT(/T44,'     SILICATE SYSTEM IN ',A38//T50,'SOLID = ',A27)
  501      586  FORMAT(/,F5.2,1P5D13.4,D14.4,D15.4,D13.4,D15.4)
  502      5861 FORMAT(/,F5.2,  5F13.4,F14.4,F15.4,F13.4,F15.4)
  503      587  FORMAT(/,F5.2,1P2D13.4,2D15.4,3D13.4)
  504      5871 FORMAT(/,F5.2,  2F13.4,2F15.4,3F13.4)
  505      5872 FORMAT(F5.2,14F10.4)
  506      5873 FORMAT(F5.2,14F6.1)
  507      129  FORMAT(/'  pH     SiO4(-4)     HSiO4(-3)  ',
  508          & ' H2SiO4(-2)    H3SiO4(-)    H4SiO4(0)   H2(SiO4)2(-6)  ',
  509          & ' MgSiO4(-2)   MgHSiO4(-)   Mg(HSiO4)2(-4)')
  510      130  FORMAT(/'  pH    CaSiO4(-2)   CaHSiO4(-)  ',
  511          & ' Ca(HSiO4)2(-4)    FeHSiO4(0)     T.D.SiO4')
  512     C
  513           RETURN
  514           END
```

```
WSIO4  Local Symbols

Name                     Class   Type              Size   Offset

AFILE . . . . . . . . . . param                           0006
```

```
WSI04  Local Symbols

Name                      Class   Type              Size   Offset

UNIT. . . . . . . . . . . . param                          000a
T1. . . . . . . . . . . . . param                          000e
TDSI04. . . . . . . . . . . param                          0012
I . . . . . . . . . . . . . local   INTEGER*4           4   01e8
J . . . . . . . . . . . . . local   INTEGER*4           4   01ec
T . . . . . . . . . . . . . local   REAL*8              8   01f0
ISTART. . . . . . . . . . . local   INTEGER*4           4   01f8
Y . . . . . . . . . . . . . local   REAL*8            112   01fc
LABEL . . . . . . . . . . . local   CHAR*38            38   026c
IEND. . . . . . . . . . . . local   INTEGER*4           4   0292
SKIP. . . . . . . . . . . . local   INTEGER*4           4   0296
PHS . . . . . . . . . . . . OUT     REAL*4            164   98b8
X . . . . . . . . . . . . . OUT     REAL*8         104632   0000
F . . . . . . . . . . . . . W       REAL*8              8   0000
F2. . . . . . . . . . . . . W       REAL*8              8   0008
F3. . . . . . . . . . . . . W       REAL*8              8   0010
NUMPH . . . . . . . . . . . W       INTEGER*4           4   0018
NPL . . . . . . . . . . . . W       INTEGER*4           4   001c
PR. . . . . . . . . . . . . W       INTEGER*4           4   003c
SLDID . . . . . . . . . . . W       CHAR*27            27   0020
```

```fortran
C
      SUBROUTINE WNH3(TDNH3,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE AMMONIA SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDNH3(41)
      REAL*8 X(319,41),Y(21)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES               '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES          '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF NITROGEN                 '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF NITROGEN        '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF NITROGEN   '
        WRITE(PR,217) LABEL,SLDID
        WRITE(PR,131)
        DO 31 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6051) PHS(I),
     &      LOGFN(X( 35,I)*F)   ,LOGFN(X( 34,I)*F),
     &      LOGFN(X( 93,I)*F)   ,LOGFN(X( 94,I)*F*F2),
     &      LOGFN(X( 95,I)*F*F3),LOGFN(X( 50,I)*F),
     &      LOGFN(X( 51,I)*F*F2),LOGFN(X( 52,I)*F*F3),
     &      LOGFN(X( 31,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6051) PHS(I),
     &      X( 35,I)/T   ,X( 34,I)/T,
     &      X( 93,I)/T   ,X( 94,I)/T*F2,
     &      X( 95,I)/T*F3,X( 50,I)/T,
     &      X( 51,I)/T*F2,X( 52,I)/T*F3,
     &      X( 31,I)/T
          ELSE
            WRITE(PR,605) PHS(I),
     &      X( 35,I)*F   ,X( 34,I)*F,
     &      X( 93,I)*F   ,X( 94,I)*F*F2,
     &      X( 95,I)*F*F3,X( 50,I)*F,
     &      X( 51,I)*F*F2,X( 52,I)*F*F3,
     &      X( 31,I)*F
          END IF
  31    CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,217) LABEL,SLDID
        WRITE(PR,132)
        DO 32 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6061) PHS(I),
     &      LOGFN(X( 32,I)*F*F2),LOGFN(X( 33,I)*F*F3),
     &      LOGFN(X(119,I)*F)   ,LOGFN(X(120,I)*F*F2),
     &      LOGFN(X(121,I)*F*F3),LOGFN(X(137,I)*F),
     &      LOGFN(X(138,I)*F*F2),LOGFN(X(139,I)*F*F3),
     &      LOGFN(X(231,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6061) PHS(I),
     &      X( 32,I)/T*F2,X( 33,I)/T*F3,
     &      X(119,I)/T   ,X(120,I)/T*F2,
     &      X(121,I)/T*F3,X(137,I)/T.
     &      X(138,I)/T*F2,X(139,I)/T*F3,
     &      X(231,I)/T
```

```
          ELSE
            WRITE(PR,606) PHS(I),
     &      X( 32,I)*F*F2,X( 33,I)*F*F3,
     &      X(119,I)*F   ,X(120,I)*F*F2,
     &      X(121,I)*F*F3,X(137,I)*F,
     &      X(138,I)*F*F2,X(139,I)*F*F3,
     &      X(231,I)*F
          END IF
   32   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,217) LABEL,SLDID
        WRITE(PR,133)
        DO 33 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6061) PHS(I),
     &      LOGFN(X(232,I)*F*F2),LOGFN(X(233,I)*F*F3),
     &      LOGFN(TDNH3(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6061) PHS(I),
     &      X(232,I)/T*F2,X(233,I)/T*F3,
     &      TDNH3(I)/T
          ELSE
            WRITE(PR,606) PHS(I),
     &      X(232,I)*F*F2,X(233,I)*F*F3,
     &      TDNH3(I)*F
          END IF
   33   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &                RECL=220)
        WRITE(7,216) LABEL
        DO 79 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X( 35,I)*F)
            Y(2 )=LOGFN(X( 34,I)*F)
            Y(3 )=LOGFN(X( 93,I)*F)
            Y(4 )=LOGFN(X( 94,I)*F*F2)
            Y(5 )=LOGFN(X( 95,I)*F*F3)
            Y(6 )=LOGFN(X( 50,I)*F)
            Y(7 )=LOGFN(X( 51,I)*F*F2)
            Y(8 )=LOGFN(X( 52,I)*F*F3)
            Y(9 )=LOGFN(X( 31,I)*F)
            Y(10)=LOGFN(X( 32,I)*F*F2)
            Y(11)=LOGFN(X( 33,I)*F*F3)
            Y(12)=LOGFN(X(119,I)*F)
            Y(13)=LOGFN(X(120,I)*F*F2)
            Y(14)=LOGFN(X(121,I)*F*F3)
            Y(15)=LOGFN(X(137,I)*F)
            Y(16)=LOGFN(X(138,I)*F*F2)
            Y(17)=LOGFN(X(139,I)*F*F3)
            Y(18)=LOGFN(X(231,I)*F)
            Y(19)=LOGFN(X(232,I)*F*F2)
            Y(20)=LOGFN(X(233,I)*F*F3)
            Y(21)=LOGFN(TDNH3(I)*F)
            WRITE(7,6062) PHS(I),(Y(J),J=1,21)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6063) PHS(I),
     &      X( 35,I)/T   ,X( 34,I)/T,
     &      X( 93,I)/T   ,X( 94,I)/T*F2,
     &      X( 95,I)/T*F3,X( 50,I)/T,
     &      X( 51,I)/T*F2,X( 52,I)/T*F3,
     &      X( 31,I)/T,
     &      X( 32,I)/T*F2,X( 33,I)/T*F3,
     &      X(119,I)/T   ,X(120,I)/T*F2,
     &      X(121,I)/T*F3,X(137,I)/T,
     &      X(138,I)/T*F2,X(139,I)/T*F3,
     &      X(231,I)/T,
     &      X(232,I)/T*F2,X(233,I)/T*F3,
     &      TDNH3(I)/T
          ELSE
            WRITE(7,6062) PHS(I),
     &      X( 35,I)*F   ,X( 34,I)*F,
     &      X( 93,I)*F   ,X( 94,I)*F*F2,
     &      X( 95,I)*F*F3,X( 50,I)*F,
     &      X( 51,I)*F*F2,X( 52,I)*F*F3,
     &      X( 31,I)*F,
     &      X( 32,I)*F*F2,X( 33,I)*F*F3,
     &      X(119,I)*F   ,X(120,I)*F*F2,
     &      X(121,I)*F*F3,X(137,I)*F,
     &      X(138,I)*F*F2,X(139,I)*F*F3,
     &      X(231,I)*F,
     &      X(232,I)*F*F2,X(233,I)*F*F3,
     &      TDNH3(I)*F
          END IF
   79   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 216  FORMAT(     '    AMMONIA SYSTEM IN ',A38)
 217  FORMAT(/T44,'    AMMONIA SYSTEM IN ',A38//T50,'SOLID = ',A27)
 605  FORMAT(/,F5.2,1P2D13.4,2(D13.4,2D15.4),D14.4)
 6051 FORMAT(/,F5.2,  2F13.4,2(F13.4,2F15.4),F14.4)
 606  FORMAT(/,F5.2,1P9D14.4)
 6061 FORMAT(/,F5.2,  9F14.4)
 6062 FORMAT(F5.2,21F10.4)
 6063 FORMAT(F5.2,21F6.1)
 Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

   693   131  FORMAT(/'  pH       NH3(0)        NH4(+)     ',
   694        & ' Hg(NH3)(+2)   Hg(NH3)2(+2)   Hg(NH3)3(+2)  Cd(NH3)(+2)   ',
   695        & ' Cd(NH3)2(+2)   Cd(NH3)3(+2)   Cu(NH3)(+2)')
   696   132  FORMAT(/'  pH    Cu(NH3)2(+2)  Cu(NH3)3(+2)   ',
   697        & ' Mn(NH3)(+2)  Mn(NH3)2(+2)  Mn(NH3)3(+2)    Zn(NH3)(+2) ',
   698        & ' Zn(NH3)2(+2)  Zn(NH3)3(+2)   Co(NH3)(+2)')
   699   133  FORMAT(/'  pH    Co(NH3)2(+2)  Co(NH3)3(+2)      ',
   700        & ' T.D.NH3')
   701   C
   702          RETURN
   703          END


WNH3  Local Symbols

Name                          Class   Type               Size   Offset

AFILE . . . . . . . . . . . . param                               0006
UNIT. . . . . . . . . . . . . param                               000a
T1. . . . . . . . . . . . . . param                               000e
TDNH3 . . . . . . . . . . . . param                               0012
I . . . . . . . . . . . . . . local   INTEGER*4             4     029a
J . . . . . . . . . . . . . . local   INTEGER*4             4     029e
T . . . . . . . . . . . . . . local   REAL*8                8     02a2
ISTART. . . . . . . . . . . . local   INTEGER*4             4     02aa
Y . . . . . . . . . . . . . . local   REAL*8              168     02ae
LABEL . . . . . . . . . . . . local   CHAR*38              38     0356
IEND. . . . . . . . . . . . . local   INTEGER*4             4     037c
SKIP. . . . . . . . . . . . . local   INTEGER*4             4     0380
PHS . . . . . . . . . . . . . OUT     REAL*4              164     98b8
X . . . . . . . . . . . . . . OUT     REAL*8           104632     0000
F . . . . . . . . . . . . . . W       REAL*8                8     0000
F2. . . . . . . . . . . . . . W       REAL*8                8     0008
F3. . . . . . . . . . . . . . W       REAL*8                8     0010
NUMPH . . . . . . . . . . . . W       INTEGER*4             4     0018
NPL . . . . . . . . . . . . . W       INTEGER*4             4     001c
PR. . . . . . . . . . . . . . W       INTEGER*4             4     003c
SLDID . . . . . . . . . . . . W       CHAR*27              27     0020


Global Symbols

Name                          Class   Type               Size   Offset

LOGFN . . . . . . . . . . . . extern  REAL*8              ***      ***
OUT . . . . . . . . . . . . . common  ***              104796     0000
W . . . . . . . . . . . . . . common  ***                  64     0000
WCL . . . . . . . . . . . . . FSUBRT  ***                 ***     0000
WFOOT . . . . . . . . . . . . extern  ***                 ***      ***
WHEAD . . . . . . . . . . . . extern  ***                 ***      ***
WNH3. . . . . . . . . . . . . FSUBRT  ***                 ***     74f4
```

```
Global Symbols

Name                         Class   Type                Size   Offset

WOC . . . . . . . . . . . . . FSUBRT  ***                 ***     2cde
WSE . . . . . . . . . . . . . FSUBRT  ***                 ***     3def
WSIO4 . . . . . . . . . . . . FSUBRT  ***                 ***     5415

Code size = a768 (42856)
Data size = 095b (2395)
Bss size  = 0384 (900)

No errors detected
```

```
      SUBROUTINE WPO4(TDPO4,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE PHOSPHATE SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDPO4(41)
      REAL*8 X(319,41),Y(23)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF PHOSPHORUS           '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF PHOSPHORUS      '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF PHOSPHORUS '
        WRITE(PR,219) LABEL,SLDID
        WRITE(PR,134)
        DO 34 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6161) PHS(I),
     &      LOGFN(X( 62,I)*F),LOGFN(X( 55,I)*F),
     &      LOGFN(X( 54,I)*F),LOGFN(X( 53,I)*F),
     &      LOGFN(X(142,I)*F),LOGFN(X( 67,I)*F),
     &      LOGFN(X( 68,I)*F),LOGFN(X( 69,I)*F),
     &      LOGFN(X( 64,I)*F),LOGFN(X( 65,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6161) PHS(I),
     &      X( 62,I)/T,X( 55,I)/T,
     &      X( 54,I)/T,X( 53,I)/T,
     &      X(142,I)/T,X( 67,I)/T,
     &      X( 68,I)/T,X( 69,I)/T,
     &      X( 64,I)/T,X( 65,I)/T
          ELSE
            WRITE(PR,616) PHS(I),
     &      X( 62,I)*F,X( 55,I)*F,
     &      X( 54,I)*F,X( 53,I)*F,
     &      X(142,I)*F,X( 67,I)*F,
     &      X( 68,I)*F,X( 69,I)*F,
     &      X( 64,I)*F,X( 65,I)*F
          END IF
   34   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,219) LABEL,SLDID
        WRITE(PR,135)
        DO 35 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6171) PHS(I),
     &      LOGFN(X( 66,I)*F),LOGFN(X(156,I)*F),
     &      LOGFN(X(157,I)*F),LOGFN(X( 77,I)*F),
     &      LOGFN(X( 78,I)*F),LOGFN(X(126,I)*F),
     &      LOGFN(X(127,I)*F),LOGFN(X(146,I)*F),
     &      LOGFN(X(234,I)*F),LOGFN(X(302,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6171) PHS(I),
     &      X( 66,I)/T,X(156,I)/T,
     &      X(157,I)/T,X( 77,I)/T,
     &      X( 78,I)/T,X(126,I)/T,
     &      X(127,I)/T,X(146,I)/T,
     &      X(234,I)/T,X(302,I)/T
          ELSE
            WRITE(PR,617) PHS(I),
     &      X( 66,I)*F,X(156,I)*F,
     &      X(157,I)*F,X( 77,I)*F,
     &      X( 78,I)*F,X(126,I)*F,
     &      X(127,I)*F,X(146,I)*F,
     &      X(234,I)*F,X(302,I)*F
          END IF
   35   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,219) LABEL,SLDID
        WRITE(PR,1351)
        DO I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6171) PHS(I),
     &      LOGFN(X(303,I)*F*F2),LOGFN(X(304,I)*F)*F3,
     &      LOGFN(TDPO4(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6171) PHS(I),
     &      X(303,I)*F2/T,X(304,I)*F3/T,
     &      TDPO4(I)/T
          ELSE
            WRITE(PR,617) PHS(I),
     &      X(303,I)*F*F2,X(304,I)*F*F3,
     &      TDPO4(I)*F
```

```
          END IF
        END DO
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &              RECL=210)
        WRITE(7,218) LABEL
        DO 80 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X( 62,I)*F)
            Y(2 )=LOGFN(X( 55,I)*F)
            Y(3 )=LOGFN(X( 54,I)*F)
            Y(4 )=LOGFN(X( 53,I)*F)
            Y(5 )=LOGFN(X(142,I)*F)
            Y(6 )=LOGFN(X( 67,I)*F)
            Y(7 )=LOGFN(X( 68,I)*F)
            Y(8 )=LOGFN(X( 69,I)*F)
            Y(9 )=LOGFN(X( 64,I)*F)
            Y(10)=LOGFN(X( 65,I)*F)
            Y(11)=LOGFN(X( 66,I)*F)
            Y(12)=LOGFN(X(156,I)*F)
            Y(13)=LOGFN(X(157,I)*F)
            Y(14)=LOGFN(X( 77,I)*F)
            Y(15)=LOGFN(X( 78,I)*F)
            Y(16)=LOGFN(X(126,I)*F)
            Y(17)=LOGFN(X(127,I)*F)
            Y(18)=LOGFN(X(146,I)*F)
            Y(19)=LOGFN(X(234,I)*F)
            Y(20)=LOGFN(X(302,I)*F)
            Y(21)=LOGFN(X(303,I)*F)*F2
            Y(22)=LOGFN(X(304,I)*F)*F3
            Y(23)=LOGFN(TDPO4(I)*F)
            WRITE(7,6173) PHS(I),(Y(J),J=1,23)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6173) PHS(I),
     &      X( 62,I)/T,X( 55,I)/T,
     &      X( 54,I)/T,X( 53,I)/T,
     &      X(142,I)/T,X( 67,I)/T,
     &      X( 68,I)/T,X( 69,I)/T,
     &      X( 64,I)/T,X( 65,I)/T,
     &      X( 66,I)/T,X(156,I)/T,
     &      X(157,I)/T,X( 77,I)/T,
     &      X( 78,I)/T,X(126,I)/T,
     &      X(127,I)/T,X(146,I)/T,
     &      X(234,I)/T,X(302,I)/T,
     &      X(303,I*F2/T),X(304,I)*F3/T,
     &      TDPO4(I)/T
          ELSE
```

```
                 WRITE(7,6172) PHS(I),
     &           X( 62,I)*F,X( 55,I)*F,
     &           X( 54,I)*F,X( 53,I)*F,
     &           X(142,I)*F,X( 67,I)*F,
     &           X( 68,I)*F,X( 69,I)*F,
     &           X( 64,I)*F,X( 65,I)*F,
     &           X( 66,I)*F,X(156,I)*F,
     &           X(157,I)*F,X( 77,I)*F,
     &           X( 78,I)*F,X(126,I)*F,
     &           X(127,I)*F,X(146,I)*F,
     &           X(234,I)*F,X(302,I)*F,
     &           X(303,I)*F*F2,X(304,I)*F*F3,
     &           TDPO4(I)*F
               END IF
   80        CONTINUE
             CLOSE (UNIT=7)
           END IF
C
 218   FORMAT(      '   PHOSPHATE SYSTEM IN '.A38)
 219   FORMAT(/T44,'   PHOSPHATE SYSTEM IN  ,A38//T50,'SOLID = ',A27)
 616   FORMAT(/,F5.2,1PD13.4,3D12.4,6D13.4)
 6161  FORMAT(/,F5.2,  F13.4,3F12.4,6F13.4)
 617   FORMAT(/,F5.2,1P4D13.4,3(D13.4,D12.4))
 6171  FORMAT(/,F5.2,  4F13.4,3(F13.4,F12.4))
 6172  FORMAT(F5.2,23F10.4)
 6173  FORMAT(F5.2,23F6.1)
 6174  FORMAT(/,F5.2,1P2D15.4,D12.4)
 6175  FORMAT(/,F5.2,  2F15.4,D12.4)
 134   FORMAT(/'  pH      PO4(-3)     HPO4(-2)    ',
     & ' H2PO4(-)    H3PO4(0)     NaHPO4(-)    MgPO4(-)     ',
     & ' MgHPO4(0)   MgH2PO4(+)    CaPO4(-)     CaHPO4(0)')
 135   FORMAT(/'  pH    CaH2PO4(+)    FeHPO4(+)   ',
     & ' FeH2PO4(+2)   FeHPO4(0)   FeH2PO4(+)   ZnHPO4(0)   ',
     & ' ZnH2PO4(+)   KHPO4(-)      CoHPO4(0)  AlH2PO4(+2)')
 1351  FORMAT(/'  pH  Al(H2PO4)2(+)  Al(H2PO4)3(0)   T.D.PO4')
C
       RETURN
       END
```

```
WPO4  Local Symbols

Name                       Class   Type             Size   Offset

AFILE . . . . . . . . . . . param                           0006
UNIT. . . . . . . . . . . . param                           000a
T1. . . . . . . . . . . . . param                           000e
TDPO4 . . . . . . . . . . . param                           0012
I . . . . . . . . . . . . . local   INTEGER*4           4   0002
J . . . . . . . . . . . . . local   INTEGER*4           4   0006
T . . . . . . . . . . . . . local   REAL*8              8   000a
```

```
WP04  Local Symbols

Name                        Class   Type              Size   Offset

ISTART. . . . . . . . . . . local   INTEGER*4            4    0012
Y . . . . . . . . . . . . . local   REAL*8             184    0016
LABEL . . . . . . . . . . . local   CHAR*38             38    00ce
IEND. . . . . . . . . . . . local   INTEGER*4            4    00f4
SKIP. . . . . . . . . . . . local   INTEGER*4            4    00f8
PHS . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . W       CHAR*27             27    0020
```

```
C
      SUBROUTINE WAC(TDAC,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE ACETATE SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDAC(41)
      REAL*8 X(319,41),Y(22)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CARBON               '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CARBON          '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CARBON     '
        WRITE(PR,221) LABEL,SLDID
        WRITE(PR,136)
        DO 36 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
```

```
            WRITE(PR,6181) PHS(I),
     &      LOGFN(X( 38,I)*F)   ,LOGFN(X( 39,I)*F),
     &      LOGFN(X(144,I)*F)   ,LOGFN(X(163,I)*F),
     &      LOGFN(X(164,I)*F*F2),LOGFN(X(165,I)*F*F3),
     &      LOGFN(X( 96,I)*F)   ,LOGFN(X( 97,I)*F*F2),
     &      LOGFN(X( 98,I)*F*F3),LOGFN(X( 56,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6181) PHS(I),
     &      X( 38,I)/T   ,X( 39,I)/T,
     &      X(144,I)/T   ,X(163,I)/T,
     &      X(164,I)/T*F2,X(165,I)/T*F3,
     &      X( 96,I)/T   ,X( 97,I)/T*F2,
     &      X( 98,I)/T*F3,X( 56,I)/T
          ELSE
            WRITE(PR,618) PHS(I),
     &      X( 38,I)*F   ,X( 39,I)*F,
     &      X(144,I)*F   ,X(163,I)*F,
     &      X(164,I)*F*F2,X(165,I)*F*F3,
     &      X( 96,I)*F   ,X( 97,I)*F*F2,
     &      X( 98,I)*F*F3,X( 56,I)*F
          END IF
   36   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,221) LABEL,SLDID
        WRITE(PR,137)
        DO 37 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6181) PHS(I),
     &      LOGFN(X( 57,I)*F*F2),LOGFN(X( 58,I)*F*F3),
     &      LOGFN(X( 36,I)*F)   ,LOGFN(X( 37,I)*F*F2),
     &      LOGFN(X( 79,I)*F)   ,LOGFN(X(118,I)*F),
     &      LOGFN(X(134,I)*F)   ,LOGFN(X(135,I)*F*F2),
     &      LOGFN(X(136,I)*F*F3),LOGFN(X(235,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6181) PHS(I),
     &      X( 57,I)/T*F2,X( 58,I)/T*F3,
     &      X( 36,I)/T   ,X( 37,I)/T*F2,
     &      X( 79,I)/T   ,X(118,I)/T,
     &      X(134,I)/T   ,X(135,I)/T*F2,
     &      X(136,I)/T*F3,X(235,I)/T
          ELSE
            WRITE(PR,618) PHS(I),
     &      X( 57,I)*F*F2,X( 58,I)*F*F3,
     &      X( 36,I)*F   ,X( 37,I)*F*F2,
     &      X( 79,I)*F   ,X(118,I)*F,
     &      X(134,I)*F   ,X(135,I)*F*F2,
     &      X(136,I)*F*F3,X(235,I)*F
          END IF
   37   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,221) LABEL,SLDID
        WRITE(PR,138)
        DO 38 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6181) PHS(I),
     &      LOGFN(X(298,I)*F),LOGFN(TDAC(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6181) PHS(I),
     &      X(298,I)/T,TDAC(I)/T
          ELSE
            WRITE(PR,618) PHS(I),
     &      X(298,I)*F,TDAC(I)*F
          END IF
   38   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=230)
        WRITE(7,220) LABEL
        DO 81 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X( 38,I)*F)
            Y(2 )=LOGFN(X( 39,I)*F)
            Y(3 )=LOGFN(X(144,I)*F)
            Y(4 )=LOGFN(X(163,I)*F)
            Y(5 )=LOGFN(X(164,I)*F*F2)
            Y(6 )=LOGFN(X(165,I)*F*F3)
            Y(7 )=LOGFN(X( 96,I)*F)
            Y(8 )=LOGFN(X( 97,I)*F*F2)
            Y(9 )=LOGFN(X( 98,I)*F*F3)
            Y(10)=LOGFN(X( 56,I)*F)
            Y(11)=LOGFN(X( 57,I)*F*F2)
            Y(12)=LOGFN(X( 58,I)*F*F3)
            Y(13)=LOGFN(X( 36,I)*F)
            Y(14)=LOGFN(X( 37,I)*F*F2)
            Y(15)=LOGFN(X( 79,I)*F)
            Y(16)=LOGFN(X(118,I)*F)
            Y(17)=LOGFN(X(134,I)*F)
            Y(18)=LOGFN(X(135,I)*F*F2)
            Y(19)=LOGFN(X(136,I)*F*F3)
            Y(20)=LOGFN(X(235,I)*F)
            Y(21)=LOGFN(X(298,I)*F)
            Y(22)=LOGFN(TDAC(I)*F)
            WRITE(7,6182) PHS(I),(Y(J),J=1,22)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6183) PHS(I),
     &      X( 38,I)/T   ,X( 39,I)/T,
     &      X(144,I)/T   ,X(163,I)/T,
     &      X(164,I)/T*F2,X(165,I)/T*F3,
     &      X( 96,I)/T   ,X( 97,I)/T*F2,
     &      X( 98,I)/T*F3,X( 56,I)/T,
     &      X( 57,I)/T*F2,X( 58,I)/T*F3,
     &      X( 36,I)/T   ,X( 37,I)/T*F2,
     &      X( 79,I)/T   ,X(118,I)/T,
     &      X(134,I)/T   ,X(135,I)/T*F2,
     &      X(136,I)/T*F3,X(235,I)/T,
     &      X(298,I)/T,TDAC(I)/T
          ELSE
            WRITE(7,6182) PHS(I),
     &      X( 38,I)*F   ,X( 39,I)*F,
     &      X(144,I)*F   ,X(163,I)*F,
     &      X(164,I)*F*F2,X(165,I)*F*F3,
     &      X( 96,I)*F   ,X( 97,I)*F*F2,
     &      X( 98,I)*F*F3,X( 56,I)*F,
     &      X( 57,I)*F*F2,X( 58,I)*F*F3,
     &      X( 36,I)*F   ,X( 37,I)*F*F2,
     &      X( 79,I)*F   ,X(118,I)*F,
     &      X(134,I)*F   ,X(135,I)*F*F2,
     &      X(136,I)*F*F3,X(235,I)*F,
     &      X(298,I)*F   ,TDAC(I)*F
          END IF
   81   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 220  FORMAT(     '    ACETATE SYSTEM IN ',A38)
 221  FORMAT(/T44,'    ACETATE SYSTEM IN ',A38//T50,'SOLID = ',A27)
 618  FORMAT(/,F5.2,1PD12.4,3(D12.4,2D13.4))
6181  FORMAT(/,F5.2,  F12.4,3(F12.4,2F13.4))
6182  FORMAT(F5.2,22F10.4)
6183  FORMAT(F5.2,22F6.1)
 136  FORMAT(/'  pH        Ac(-)     HAc(0)          ',
     & ' NaAc(0)     FeAc(+2)    FeAc2(+)     FeAc3(0)     ',
     & ' HgAc(+)    HgAc2(0)    HgAc3(-)     CdAc(+)')
 137  FORMAT(/'  pH     CdAc2(0)    CdAc3(-)     ',
     & ' CuAc(+)     CuAc2(0)     FeAc(+)     MnAc(+)      ',
     & ' ZnAc(+)    ZnAc2(0)    ZnAc3(-)     CoAc(+)')
 138  FORMAT(/'  pH      AlAc(+2)    T.D.Ac')
C
      RETURN
      END
```

```
WAC  Local Symbols

Name                          Class    Type              Size   Offset

AFILE . . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . . param                             000e
TDAC. . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local    INTEGER*4            4   00fc
J . . . . . . . . . . . . . . local    INTEGER*4            4   0100
T . . . . . . . . . . . . . . local    REAL*8               8   0104
ISTART. . . . . . . . . . . . local    INTEGER*4            4   010c
Y . . . . . . . . . . . . . . local    REAL*8             176   0110
LABEL . . . . . . . . . . . . local    CHAR*38             38   01c0
IEND. . . . . . . . . . . . . local    INTEGER*4            4   01e6
SKIP. . . . . . . . . . . . . local    INTEGER*4            4   01ea
PHS . . . . . . . . . . . . . OUT      REAL*4             164   98b8
X . . . . . . . . . . . . . . OUT      REAL*8          104632   0000
F . . . . . . . . . . . . . . W        REAL*8               8   0000
F2. . . . . . . . . . . . . . W        REAL*8               8   0008
F3. . . . . . . . . . . . . . W        REAL*8               8   0010
NUMPH . . . . . . . . . . . . W        INTEGER*4            4   0018
NPL . . . . . . . . . . . . . W        INTEGER*4            4   001c
PR. . . . . . . . . . . . . . W        INTEGER*4            4   003c
SLDID . . . . . . . . . . . . W        CHAR*27             27   0020

    374  C
    375        SUBROUTINE WS(TDS,T1,UNIT,AFILE)
    376  C     THIS BLOCK WRITES THE SULPHIDE SUBSYSTEM AQUEOUS SPECIES REPORT.
    377  C
    378        REAL*4 PHS(41),TDS(41)
    379        REAL*8 X(319,41),Y(10)
    380        COMMON /OUT/X,PHS
    381        INTEGER I,J,ISTART,IEND,SKIP,UNIT
    382        REAL*8 F,F2,F3,T,T1
    383        INTEGER NUMPH,NPL,PR
    384        CHARACTER LABEL*38,SLDID*27
    385        CHARACTER AFILE*15
    386        REAL*8 LOGFN
    387        COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
    388  C
    389        ISTART=1
    390        IEND=0
    391        DO WHILE(IEND.NE.NUMPH)
    392          IEND=MIN0(IEND+NPL,NUMPH)
    393  C
    394          CALL WHEAD
    395          IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES              '
    396          IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES         '
    397          IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
    398          IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF SULPHUR                 '
```

```
399          IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF SULPHUR             '
400          IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF SULPHUR        '
401          WRITE(PR,223) LABEL,SLDID
402          WRITE(PR,139)
403          DO 39 I=ISTART,IEND
404            IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
405              WRITE(PR,6321) PHS(I),
406       &      LOGFN(X( 61,I)*F)   ,LOGFN(X( 60,I)*F),
407       &      LOGFN(X( 59,I)*F)   ,LOGFN(X( 99,I)*F*F2),
408       &      LOGFN(X( 81,I)*F)   ,LOGFN(X( 82,I)*F*F2),
409       &      LOGFN(X( 83,I)*F*F3),LOGFN(X(188,I)*F*F2),
410       &      LOGFN(X(140,I)*F)   ,LOGFN(TDS(I)*F)
411            ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
412              T=100
413              IF (T1.NE.0D0) T=T1/100
414              WRITE(PR,6321) PHS(I),
415       &      X( 61,I)/T   ,X( 60,I)/T,
416       &      X( 59,I)/T   ,X( 99,I)/T*F2,
417       &      X( 81,I)/T   ,X( 82,I)/T*F2,
418       &      X( 83,I)/T*F3,X(188,I)/T*F2,
419       &      X(140,I)/T   ,TDS(I)/T
420            ELSE
421              WRITE(PR,632) PHS(I),
422       &      X( 61,I)*F   ,X( 60,I)*F,
423       &      X( 59,I)*F   ,X( 99,I)*F*F2,
424       &      X( 81,I)*F   ,X( 82,I)*F*F2,
425       &      X( 83,I)*F*F3,X(188,I)*F*F2,
426       &      X(140,I)*F   ,TDS(I)*F
427            END IF
428     39   CONTINUE
429          SKIP=2*(NPL+ISTART-IEND)
430          CALL WFOOT(SKIP)
431  C
432          ISTART=IEND+1
433        END DO
434        IF (AFILE(1:1).NE.' ') THEN
435          OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
436       &                RECL=110)
437          WRITE(7,222) LABEL
438          DO 82 I=1,NUMPH
439            IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
440              Y(1 )=LOGFN(X( 61,I)*F)
441              Y(2 )=LOGFN(X( 60,I)*F)
442              Y(3 )=LOGFN(X( 59,I)*F)
443              Y(4 )=LOGFN(X( 99,I)*F*F2)
444              Y(5 )=LOGFN(X( 81,I)*F)
445              Y(6 )=LOGFN(X( 82,I)*F*F2)
446              Y(7 )=LOGFN(X( 83,I)*F*F3)
447              Y(8 )=LOGFN(X(188,I)*F*F2)
448              Y(9 )=LOGFN(X(140,I)*F)
449              Y(10)=LOGFN(TDS(I)*F)
```

```
            WRITE(7,6322) PHS(I),(Y(J),J=1,10)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6323) PHS(I),
     &      X( 61,I)/T   ,X( 60,I)/T,
     &      X( 59,I)/T   ,X( 99,I)/T*F2,
     &      X( 81,I)/T   ,X( 82,I)/T*F2,
     &      X( 83,I)/T*F3,X(188,I)/T*F2,
     &      X(140,I)/T   ,TDS(I)/T
          ELSE
            WRITE(7,6322) PHS(I),
     &      X( 61,I)*F   ,X( 60,I)*F,
     &      X( 59,I)*F   ,X( 99,I)*F*F2,
     &      X( 81,I)*F   ,X( 82,I)*F*F2,
     &      X( 83,I)*F*F3,X(188,I)*F*F2,
     &      X(140,I)*F   ,TDS(I)*F
          END IF
   82   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 222  FORMAT(     '    SULPHIDE SYSTEM IN ',A38)
 223  FORMAT(/T44,'    SULPHIDE SYSTEM IN ',A38//T50,'SOLID = ',A27)
 632  FORMAT(/,F5.2,1PD12.4,3(D12.4,2D13.4))
 6321 FORMAT(/,F5.2,  F12.4,3(F12.4,2F13.4))
 6322 FORMAT(F5.2,10F10.4)
 6323 FORMAT(F5.2,10F6.1)
 139  FORMAT(/' pH       S(-2)      HS(-)      ',
     & ' H2S(0)     Hg(HS)2(0)    CdHS(+)    Cd(HS)2(0)  ',
     & ' Cd(HS)3(-)  Zn(HS)2(0)    ZnHSOH(0)      T.D.S')
C
      RETURN
      END
```

```
WS  Local Symbols

Name                         Class   Type              Size   Offset

AFILE . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . param                             000e
TDS . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . local   INTEGER*4            4    01ee
J . . . . . . . . . . . . . local   INTEGER*4            4    01f2
T . . . . . . . . . . . . . local   REAL*8               8    01f6
ISTART. . . . . . . . . . . local   INTEGER*4            4    01fe
Y . . . . . . . . . . . . . local   REAL*8              80    0202
LABEL . . . . . . . . . . . local   CHAR*38             38    0252
IEND. . . . . . . . . . . . local   INTEGER*4            4    0278
SKIP. . . . . . . . . . . . local   INTEGER*4            4    027c
PHS . . . . . . . . . . . . OUT     REAL*4             164    98b8
```

```
WS  Local Symbols

Name                           Class   Type             Size   Offset

X . . . . . . . . . . . . . . .OUT     REAL*8         104632    0000
F . . . . . . . . . . . . . . .W       REAL*8              8    0000
F2. . . . . . . . . . . . . . .W       REAL*8              8    0008
F3. . . . . . . . . . . . . . .W       REAL*8              8    0010
NUMPH . . . . . . . . . . . . .W       INTEGER*4           4    0018
NPL . . . . . . . . . . . . . .W       INTEGER*4           4    001c
PR. . . . . . . . . . . . . . .W       INTEGER*4           4    003c
SLDID . . . . . . . . . . . . .W       CHAR*27            27    0020
```

```
C
      SUBROUTINE WSS(TDSS,T1,UNIT,AFILE)
C     THIS BLOCK WRITES THE SEDIMENT SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDSS(41)
      REAL*8 X(319,41),Y(14)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF METAL             '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF METAL        '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF METAL   '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF METAL                '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF METAL           '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF METAL      '
        WRITE(PR,225) LABEL
        WRITE(PR,168)
        DO 68 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6331) PHS(I),
     &                          LOGFN(X(271,I)*F),
     &      LOGFN(X(272,I)*F),LOGFN(X(273,I)*F),
     &      LOGFN(X(274,I)*F),LOGFN(X(275,I)*F),
     &      LOGFN(X(276,I)*F),LOGFN(X(277,I)*F),
     &      LOGFN(X(278,I)*F)
```

```
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (TDSS(I).NE.0D0) T=TDSS(I)/100
            WRITE(PR,6331) PHS(I),
     &                 X(271,I)/T,
     &      X(272,I)/T,X(273,I)/T,
     &      X(274,I)/T,X(275,I)/T,
     &      X(276,I)/T,X(277,I)/T,
     &      X(278,I)/T
          ELSE
            WRITE(PR,633) PHS(I),
     &                 X(271,I)*F,
     &      X(272,I)*F,X(273,I)*F,
     &      X(274,I)*F,X(275,I)*F,
     &      X(276,I)*F,X(277,I)*F,
     &      X(278,I)*F
          END IF
   68   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,225) LABEL
        WRITE(PR,169)
        DO 69 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6191) PHS(I),
     &      LOGFN(X(279,I)*F),LOGFN(X(280,I)*F),
     &      LOGFN(X(281,I)*F),LOGFN(X(282,I)*F),
     &      LOGFN(X(283,I)*F),LOGFN(TDSS(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (TDSS(I).NE.0D0) T=TDSS(I)/100
            WRITE(PR,6191) PHS(I),
     &      X(279,I)/T,X(280,I)/T,
     &      X(281,I)/T,X(282,I)/T,
     &      X(283,I)/T,TDSS(I)/T
          ELSE
            WRITE(PR,619) PHS(I),
     &      X(279,I)*F,X(280,I)*F,
     &      X(281,I)*F,X(282,I)*F,
     &      X(283,I)*F,TDSS(I)*F
          END IF
   69   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=150)
```

```
        WRITE(7,224) LABEL
        DO 83 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(271,I)*F)
            Y(2 )=LOGFN(X(272,I)*F)
            Y(3 )=LOGFN(X(273,I)*F)
            Y(4 )=LOGFN(X(274,I)*F)
            Y(5 )=LOGFN(X(275,I)*F)
            Y(6 )=LOGFN(X(276,I)*F)
            Y(7 )=LOGFN(X(277,I)*F)
            Y(8 )=LOGFN(X(278,I)*F)
            Y(9 )=LOGFN(X(279,I)*F)
            Y(10)=LOGFN(X(280,I)*F)
            Y(11)=LOGFN(X(281,I)*F)
            Y(12)=LOGFN(X(282,I)*F)
            Y(13)=LOGFN(X(283,I)*F)
            Y(14)=LOGFN(TDSS(I)*F)
            WRITE(7,6192) PHS(I),(Y(J),J=1,14)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (TDSS(I).NE.0D0) T=TDSS(I)/100
            WRITE(7,6193) PHS(I),
     &                 X(271,I)/T,
     &      X(272,I)/T,X(273,I)/T,
     &      X(274,I)/T,X(275,I)/T,
     &      X(276,I)/T,X(277,I)/T,
     &      X(278,I)/T,
     &      X(279,I)/T,X(280,I)/T,
     &      X(281,I)/T,X(282,I)/T,
     &      X(283,I)/T,TDSS(I)/T
          ELSE
            WRITE(7,6192) PHS(I),
     &                 X(271,I)*F,
     &      X(272,I)*F,X(273,I)*F,
     &      X(274,I)*F,X(275,I)*F,
     &      X(276,I)*F,X(277,I)*F,
     &      X(278,I)*F,
     &      X(279,I)*F,X(280,I)*F,
     &      X(281,I)*F,X(282,I)*F,
     &      X(283,I)*F,TDSS(I)*F
          END IF
   83   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 224  FORMAT(     'SUSPENDED SEDIMENT SYSTEM IN ',A38)
 225  FORMAT(/T39,'SUSPENDED SEDIMENT SYSTEM IN ',A38//)
 633  FORMAT(/,F5.2,'              ',1P3(D12.4,2D13.4))
 6331 FORMAT(/,F5.2,'              ',  3(F12.4,2F13.4))
 619  FORMAT(/,F5.2,1PD12.4,3(D12.4,2D13.4))
 6191 FORMAT(/,F5.2,  F12.4,3(F12.4,2F13.4))
```

```
  621   6192 FORMAT(F5.2,14F10.4)
  622   6193 FORMAT(F5.2,14F6.1)
  623   168  FORMAT(/'  pH      SSH2        SSCd   ',
  624       & '     SSCu      SSFe(II)     SSFe(III)     SSMn    ',
  625       & '     SSZn         SSK         SSCo    ')
  626   169  FORMAT(/'  pH     SSHNa  ',
  627       & '    SSMg         SSCa        SSHg(I)     SSHg(II) ',
  628       & '    T.SSMe  ')
  629   C
  630          RETURN
  631          END
***** RES3.FOR(631) : warning F4202: WSS : formal argument T1 : never used


WSS  Local Symbols

Name                          Class   Type                Size   Offset

AFILE . . . . . . . . . . . . param                               0006
UNIT. . . . . . . . . . . . . param                               000a
T1. . . . . . . . . . . . . . param                               000e
TDSS. . . . . . . . . . . . . param                               0012
I . . . . . . . . . . . . . . local   INTEGER*4              4    0280
J . . . . . . . . . . . . . . local   INTEGER*4              4    0284
T . . . . . . . . . . . . . . local   REAL*8                 8    0288
ISTART. . . . . . . . . . . . local   INTEGER*4              4    0290
Y . . . . . . . . . . . . . . local   REAL*8               112    0294
LABEL . . . . . . . . . . . . local   CHAR*38               38    0304
IEND. . . . . . . . . . . . . local   INTEGER*4              4    032a
SKIP. . . . . . . . . . . . . local   INTEGER*4              4    032e
NUMPH . . . . . . . . . . . . W       INTEGER*4              4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4              4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4              4    003c
SLDID . . . . . . . . . . . . W       CHAR*27               27    0020
PHS . . . . . . . . . . . . . OUT     REAL*4               164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8            104632    0000
F . . . . . . . . . . . . . . W       REAL*8                 8    0000
F2. . . . . . . . . . . . . . W       REAL*8                 8    0008
F3. . . . . . . . . . . . . . W       REAL*8                 8    0010

  632   C
  633          SUBROUTINE WF(TDF,T1,UNIT,AFILE)
  634   C      THIS BLOCK WRITES THE FLUORIDE SUBSYSTEM AQUEOUS SPECIES REPORT.
  635   C
  636          REAL*4 PHS(41),TDF(41)
  637          REAL*8 X(319,41),Y(10)
  638          COMMON /OUT/X,PHS
  639          INTEGER I,J,ISTART,IEND,SKIP,UNIT
  640          REAL*8 F,F2,F3,T,T1
  641          INTEGER NUMPH,NPL,PR
```

```
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MINO(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF FLUORIDE             '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF FLUORIDE        '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF FLUORIDE   '
        WRITE(PR,255) LABEL,SLDID
        WRITE(PR,174)
        DO 102 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6451) PHS(I),
     &      LOGFN(X(306,I)*F)   ,LOGFN(X(307,I)*F),
     &      LOGFN(X(312,I)*F)   ,LOGFN(X(313,I)*F),
     &      LOGFN(X(317,I)*F)   ,LOGFN(X(318,I)*F*F2),
     &      LOGFN(X(319,I)*F*F3),LOGFN(X(315,I)*F),
     &      LOGFN(X(314,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6451) PHS(I),
     &      X(306,I)/T   ,X(307,I)/T,
     &      X(312,I)/T   ,X(313,I)/T,
     &      X(317,I)/T   ,X(318,I)/T*F2,
     &      X(319,I)/T*F3,X(315,I)/T,
     &      X(314,I)/T
          ELSE
            WRITE(PR,645) PHS(I),
     &      X(306,I)*F   ,X(307,I)*F,
     &      X(312,I)*F   ,X(313,I)*F,
     &      X(317,I)*F   ,X(318,I)*F*F2,
     &      X(319,I)*F*F3,X(315,I)*F,
     &      X(314,I)*F
          END IF
  102   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,255) LABEL,SLDID
        WRITE(PR,1741)
        DO I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6451) PHS(I),
     &      LOGFN(X(316,I)*F)   ,LOGFN(X(308,I)*F),
     &      LOGFN(X(309,I)*F)*F2,LOGFN(X(310,I)*F*F3),
     &      LOGFN(TDF(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6451) PHS(I),
     &      X(316,I)/T   ,X(308,I)/T,
     &      X(309,I)/T*F2,X(310,I)/T*F3,
     &      TDF(I)/T
          ELSE
            WRITE(PR,645) PHS(I),
     &      X(316,I)*F   ,X(308,I)*F,
     &      X(309,I)*F*F2,X(310,I)*F*F3,
     &      TDF(I)*F
          END IF
        END DO
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &           RECL=110)
        WRITE(7,254) LABEL
        DO 103 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(306,I)*F)
            Y(2 )=LOGFN(X(307,I)*F)
            Y(3 )=LOGFN(X(312,I)*F)
            Y(4 )=LOGFN(X(313,I)*F)
            Y(5 )=LOGFN(X(317,I)*F)
            Y(6 )=LOGFN(X(318,I)*F*F2)
            Y(7 )=LOGFN(X(319,I)*F*F3)
            Y(8 )=LOGFN(X(315,I)*F)
            Y(9 )=LOGFN(X(314,I)*F)
            Y(10)=LOGFN(X(316,I)*F)
            Y(11)=LOGFN(X(308,I)*F)
            Y(12)=LOGFN(X(309,I)*F*F2)
            Y(13)=LOGFN(X(310,I)*F*F3)
            Y(14)=LOGFN(TDF(I)*F)
            WRITE(7,6452) PHS(I),(Y(J),J=1,14)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6453) PHS(I),
     &      X(306,I)/T   ,X(307,I)/T,
     &      X(312,I)/T   ,X(313,I)/T,
     &      X(317,I)/T   ,X(318,I)/T*F2,
  744       &      X(319,I)/T*F3,X(315,I)/T,
  745       &      X(314,I)/T,
  746       &      X(316,I)/T   ,X(308,I)/T,
  747       &      X(309,I)/T*F2,X(310,I)/T*F3,
  748       &      TDF(I)/T
  749            ELSE
  750              WRITE(7,6452) PHS(I),
  751       &      X(306,I)*F   ,X(307,I)*F,
  752       &      X(312,I)*F   ,X(313,I)*F,
  753       &      X(317,I)*F   ,X(318,I)*F*F2,
  754       &      X(319,I)*F*F3,X(315,I)*F,
  755       &      X(314,I)*F,
  756       &      X(316,I)*F   ,X(308,I)*F,
  757       &      X(309,I)*F*F2,X(310,I)*F*F3,
  758       &      TDF(I)*F
  759            END IF
  760   103    CONTINUE
  761          CLOSE (UNIT=7)
  762        END IF
  763  C
  764   254  FORMAT(     '    FLUORIDE SYSTEM IN ',A38)
  765   255  FORMAT(/T44,'    FLUORIDE SYSTEM IN ',A38//T50,'SOLID = ',A27)
  766   645  FORMAT(/,F5.2,1P9D12.4)
  767   6451 FORMAT(/,F5.2,  9F12.4)
  768   6452 FORMAT(F5.2,10F10.4)
  769   6453 FORMAT(F5.2,10F6.1)
  770   174  FORMAT(/'  pH       F(-)        HF(0)    ',
  771       & '   MgF(+)      CaF(+)      FeF(+2)     FeF2(+)  ',
  772       & '   FeF3(0)     HgF(+)      CuF(+)    ')
  773   1741 FORMAT(/'  pH    ZnF(+)      AlF(+2)     AlF2(+)  ',
  774       & '   AlF3(0)      T.D.F    ')
  775  C
  776        RETURN
  777        END
```

```
WF  Local Symbols

Name                           Class   Type              Size   Offset

AFILE . . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . . param                             000e
TDF . . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local   INTEGER*4            4    0332
J . . . . . . . . . . . . . . local   INTEGER*4            4    0336
T . . . . . . . . . . . . . . local   REAL*8               8    033a
ISTART. . . . . . . . . . . . local   INTEGER*4            4    0342
Y . . . . . . . . . . . . . . local   REAL*8              80    0346
LABEL . . . . . . . . . . . . local   CHAR*38             38    0396
IEND. . . . . . . . . . . . . local   INTEGER*4            4    03bc
```

```
WF  Local Symbols

Name                         Class   Type              Size   Offset

SKIP. . . . . . . . . . . . . local   INTEGER*4            4    03c0
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020


Global Symbols

Name                         Class   Type              Size   Offset

LOGFN . . . . . . . . . . . . extern  REAL*8             ***     ***
OUT . . . . . . . . . . . . . common  ***             104796    0000
W . . . . . . . . . . . . . . common  ***                 64    0000
WAC . . . . . . . . . . . . . FSUBRT  ***                ***    347b
WF. . . . . . . . . . . . . . FSUBRT  ***                ***    a1b5
WFOOT . . . . . . . . . . . . extern  ***                ***     ***
WHEAD . . . . . . . . . . . . extern  ***                ***     ***
WP04. . . . . . . . . . . . . FSUBRT  ***                ***    0000
WS. . . . . . . . . . . . . . FSUBRT  ***                ***    68a2
WSS . . . . . . . . . . . . . FSUBRT  ***                ***    80cf

Code size = c366 (50022)
Data size = 09cb (2507)
Bss size  = 03c4 (964)

No errors detected
```

```
      SUBROUTINE WNA(TDNA,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE SODIUM SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDNA(41)
      REAL*8 X(319,41),Y(9)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF SODIUM               '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF SODIUM          '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF SODIUM     '
        WRITE(PR,227) LABEL,SLDID
        WRITE(PR,140)
        DO 40 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6341) PHS(I),
     &      LOGFN(X(181,I)*F),LOGFN(X(141,I)*F),
     &      LOGFN(X(143,I)*F),LOGFN(X(194,I)*F),
     &      LOGFN(X(240,I)*F),LOGFN(X(142,I)*F),
     &      LOGFN(X(144,I)*F),LOGFN(X(279,I)*F),
     &      LOGFN(TDNA(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6341) PHS(I),
     &      X(181,I)/T,X(141,I)/T,
     &      X(143,I)/T,X(194,I)/T,
     &      X(240,I)/T,X(142,I)/T,
     &      X(144,I)/T,X(279,I)/T,
     &      TDNA(I)/T
          ELSE
            WRITE(PR,634) PHS(I),
     &      X(181,I)*F,X(141,I)*F,
     &      X(143,I)*F,X(194,I)*F,
     &      X(240,I)*F,X(142,I)*F,
     &      X(144,I)*F,X(279,I)*F,
     &      TDNA(I)*F
          END IF
   40   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=100)
        WRITE(7,226) LABEL
        DO 84 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(181,I)*F)
            Y(2 )=LOGFN(X(141,I)*F)
            Y(3 )=LOGFN(X(143,I)*F)
            Y(4 )=LOGFN(X(194,I)*F)
            Y(5 )=LOGFN(X(240,I)*F)
            Y(6 )=LOGFN(X(142,I)*F)
            Y(7 )=LOGFN(X(144,I)*F)
            Y(8 )=LOGFN(X(279,I)*F)
            Y(9 )=LOGFN(TDNA(I)*F)
            WRITE(7,6342) PHS(I),(Y(J),J=1,9)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6343) PHS(I),
     &      X(181,I)/T,X(141,I)/T,X(143,I)/T,X(194,I)/T,
     &      X(240,I)/T,X(142,I)/T,X(144,I)/T,X(279,I)/T,
     &      TDNA(I)/T
          ELSE
            WRITE(7,6342) PHS(I),
     &      X(181,I)*F,X(141,I)*F,X(143,I)*F,X(194,I)*F,
     &      X(240,I)*F,X(142,I)*F,X(144,I)*F,X(279,I)*F,
     &      TDNA(I)*F
          END IF
   84   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 226  FORMAT(     '     SODIUM SYSTEM IN ',A38)
 227  FORMAT(/T44,'     SODIUM SYSTEM IN ',A38//T50,'SOLID = ',A27)
 634  FORMAT(/,F5.2,1PD12.4,3(D12.4,2D13.4))
 6341 FORMAT(/,F5.2,  F12.4,3(F12.4,2F13.4))
 6342 FORMAT(F5.2,9F10.4)
 6343 FORMAT(F5.2,9F6.1)
 140  FORMAT(/'  pH       Na(+)      NaOH(0)     ',
     & ' NaSO4(-)    NaEDTA(-3)   NaNTA(-2)    NaHPO4(-)    ',
     & ' NaAc(0)        SSHNa     T.D.Na')
C
      RETURN
      END
```

```
WNA  Local Symbols

Name                          Class   Type              Size   Offset

MTLFILE . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . . param                             000e
TDNA. . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local   INTEGER*4            4    0002
J . . . . . . . . . . . . . . local   INTEGER*4            4    0006
T . . . . . . . . . . . . . . local   REAL*8               8    000a
ISTART. . . . . . . . . . . . local   INTEGER*4            4    0012
Y . . . . . . . . . . . . . . local   REAL*8              72    0016
LABEL . . . . . . . . . . . . local   CHAR*38             38    005e
IEND. . . . . . . . . . . . . local   INTEGER*4            4    0084
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    0088
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020

    103  C
    104        SUBROUTINE WMG(TDMG,T1,UNIT,MTLFILE)
    105  C     THIS BLOCK WRITES THE MAGNESIUM SUBSYSTEM AQUEOUS SPECIES REPORT.
    106  C
    107        REAL*4 PHS(41),TDMG(41)
    108        REAL*8 X(319,41),Y(17)
    109        COMMON /OUT/X,PHS
    110        INTEGER I,J,ISTART,IEND,SKIP,UNIT
    111        REAL*8 F,F2,F3,T,T1
    112        INTEGER NUMPH,NPL,PR
    113        CHARACTER LABEL*38,SLDID*27
    114        CHARACTER MTLFILE*15
    115        REAL*8 LOGFN
    116        COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
    117  C
    118        ISTART=1
    119        IEND=0
    120        DO WHILE(IEND.NE.NUMPH)
    121          IEND=MIN0(IEND+NPL,NUMPH)
    122  C
    123          CALL WHEAD
    124          IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES              '
    125          IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES         '
```

```
       IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
       IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF MAGNESIUM            '
       IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF MAGNESIUM       '
       IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF MAGNESIUM  '
       WRITE(PR,229) LABEL,SLDID
       WRITE(PR,141)
       DO 41 I=ISTART,IEND
         IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
           WRITE(PR,6351) PHS(I),
     &     LOGFN(X(  5,I)*F),LOGFN(X( 14,I)*F),
     &     LOGFN(X( 10,I)*F),LOGFN(X(195,I)*F),
     &     LOGFN(X(196,I)*F),LOGFN(X(241,I)*F),
     &     LOGFN(X(  9,I)*F),LOGFN(X( 12,I)*F),
     &     LOGFN(X(173,I)*F),LOGFN(X(174,I)*F)
         ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
           T=1D0
           IF (T1.NE.0D0) T=T1/100
           WRITE(PR,6351) PHS(I),
     &     X(  5,I)/T,X( 14,I)/T,
     &     X( 10,I)/T,X(195,I)/T,
     &     X(196,I)/T,X(241,I)/T,
     &     X(  9,I)/T,X( 12,I)/T,
     &     X(173,I)/T,X(174,I)/T
         ELSE
           WRITE(PR,635) PHS(I),
     &     X(  5,I)*F,X( 14,I)*F,
     &     X( 10,I)*F,X(195,I)*F,
     &     X(196,I)*F,X(241,I)*F,
     &     X(  9,I)*F,X( 12,I)*F,
     &     X(173,I)*F,X(174,I)*F
         END IF
    41 CONTINUE
       SKIP=2*(NPL+ISTART-IEND)
       CALL WFOOT(SKIP)
C
       CALL WHEAD
       WRITE(PR,229) LABEL,SLDID
       WRITE(PR,142)
       DO 42 I=ISTART,IEND
         IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
           WRITE(PR,6361) PHS(I),
     &     LOGFN(X(175,I)*F),LOGFN(X( 67,I)*F),
     &     LOGFN(X( 68,I)*F),LOGFN(X( 69,I)*F),
     &     LOGFN(X(280,I)*F),LOGFN(X(312,I)*F),
     &     LOGFN(TDMG(I)*F)
         ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
           WRITE(PR,6361) PHS(I),
     &     X(175,I)/T,X( 67,I)/T,
     &     X( 68,I)/T,X( 69,I)/T,
     &     X(280,I)/T,X(312,I)/T,
     &     TDMG(I)/T
```

```
          ELSE
            WRITE(PR,636) PHS(I),
     &      X(175,I)*F,X( 67,I)*F,
     &      X( 68,I)*F,X( 69,I)*F,
     &      X(280,I)*F,X(312,I)*F,
     &      TDMG(I)*F
          END IF
   42   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &                RECL=180)
        WRITE(7,228) LABEL
        DO 85 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(  5,I)*F)
            Y(2 )=LOGFN(X( 14,I)*F)
            Y(3 )=LOGFN(X( 10,I)*F)
            Y(4 )=LOGFN(X(195,I)*F)
            Y(5 )=LOGFN(X(196,I)*F)
            Y(6 )=LOGFN(X(241,I)*F)
            Y(7 )=LOGFN(X(  9,I)*F)
            Y(8 )=LOGFN(X( 12,I)*F)
            Y(9 )=LOGFN(X(173,I)*F)
            Y(10)=LOGFN(X(174,I)*F)
            Y(11)=LOGFN(X(175,I)*F)
            Y(12)=LOGFN(X( 67,I)*F)
            Y(13)=LOGFN(X( 68,I)*F)
            Y(14)=LOGFN(X( 69,I)*F)
            Y(15)=LOGFN(X(280,I)*F)
            Y(16)=LOGFN(X(312,I)*F)
            Y(17)=LOGFN(TDMG(I)*F)
            WRITE(7,6362) PHS(I),(Y(J),J=1,17)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6363) PHS(I),
     &      X(  5,I)/T,X( 14,I)/T,X( 10,I)/T,X(195,I)/T,
     &      X(196,I)/T,X(241,I)/T,X(  9,I)/T,X( 12,I)/T,
     &      X(173,I)/T,X(174,I)/T,X(175,I)/T,X( 67,I)/T,
     &      X( 68,I)/T,X( 69,I)/T,X(280,I)/T,X(312,I)/T,
     &      TDMG(I)/T
          ELSE
            WRITE(7,6362) PHS(I),
     &      X(  5,I)*F,X( 14,I)*F,X( 10,I)*F,X(195,I)*F,
     &      X(196,I)*F,X(241,I)*F,X(  9,I)*F,X( 12,I)*F,
     &      X(173,I)*F,X(174,I)*F,X(175,I)*F,X( 67,I)*F,
     &      X( 68,I)*F,X( 69,I)*F,X(280,I)*F,X(312,I)*F,
     &      TDMG(I)*F
          END IF
```

```
  228      85   CONTINUE
  229           CLOSE (UNIT=7)
  230         END IF
  231  C
  232   228   FORMAT(     '   MAGNESIUM SYSTEM IN ',A38)
  233   229   FORMAT(/T44,'   MAGNESIUM SYSTEM IN ',A38//T50,'SOLID = ',A27)
  234   635   FORMAT(/,F5.2,1PD12.4,3(D12.4,2D13.4))
  235   6351  FORMAT(/,F5.2,  F12.4,3(F12.4,2F13.4))
  236   636   FORMAT(/,F5.2,1P2D15.4,4D13.4,D12.4)
  237   6361  FORMAT(/,F5.2,2F15.4,4F13.4,F12.4)
  238   6362  FORMAT(F5.2,17F10.4)
  239   6363  FORMAT(F5.2,17F6.1)
  240   141   FORMAT(/'  pH      Mg(+2)       MgOH(+)     ',
  241        & ' MgSO4(0)    MgEDTA(-2)  MgHEDTA(-)    MgNTA(-)     ',
  242        & ' MgCO3(0)    MgHCO3(+)   MgSiO4(-2)   MgHSiO4(-)')
  243   142   FORMAT(/'  pH    Mg(HSiO4)2(-4)    MgPO4(-)     ',
  244        & ' MgHPO4(0)   MgH2PO4(+)      SSMg        MgF(+)    '
  245        & '    T.D.Mg')
  246  C
  247         RETURN
  248         END


WMG  Local Symbols

Name                          Class   Type              Size   Offset

MTLFILE . . . . . . . . . . . param                            0006
UNIT. . . . . . . . . . . . . param                            000a
T1. . . . . . . . . . . . . . param                            000e
TDMG. . . . . . . . . . . . . param                            0012
I . . . . . . . . . . . . . . local   INTEGER*4            4   008c
J . . . . . . . . . . . . . . local   INTEGER*4            4   0090
T . . . . . . . . . . . . . . local   REAL*8               8   0094
ISTART. . . . . . . . . . . . local   INTEGER*4            4   009c
Y . . . . . . . . . . . . . . local   REAL*8             136   00a0
LABEL . . . . . . . . . . . . local   CHAR*38             38   0128
IEND. . . . . . . . . . . . . local   INTEGER*4            4   014e
SKIP. . . . . . . . . . . . . local   INTEGER*4            4   0152
F . . . . . . . . . . . . . . W       REAL*8               8   0000
F2. . . . . . . . . . . . . . W       REAL*8               8   0008
F3. . . . . . . . . . . . . . W       REAL*8               8   0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4   0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4   001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4   003c
SLDID . . . . . . . . . . . . W       CHAR*27             27   0020
PHS . . . . . . . . . . . . . OUT     REAL*4             164   98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632   0000

  249  C
  250         SUBROUTINE WCA(TDCA,T1,UNIT,MTLFILE)
```

```
C     THIS BLOCK WRITES THE CALCIUM SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDCA(41)
      REAL*8 X(319,41),Y(18)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CALCIUM              '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CALCIUM         '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CALCIUM    '
        WRITE(PR,231) LABEL,SLDID
        WRITE(PR,143)
        DO 43 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6111) PHS(I),
     &      LOGFN(X(  3,I)*F),LOGFN(X( 13,I)*F),
     &      LOGFN(X(  8,I)*F),LOGFN(X(197,I)*F),
     &      LOGFN(X(198,I)*F),LOGFN(X(242,I)*F),
     &      LOGFN(X(243,I)*F),LOGFN(X(  7,I)*F),
     &      LOGFN(X( 11,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6111) PHS(I),
     &      X(  3,I)/T,X( 13,I)/T,
     &      X(  8,I)/T,X(197,I)/T,
     &      X(198,I)/T,X(242,I)/T,
     &      X(243,I)/T,X(  7,I)/T,
     &      X( 11,I)/T
          ELSE
            WRITE(PR,611) PHS(I),
     &      X(  3,I)*F,X( 13,I)*F,
     &      X(  8,I)*F,X(197,I)*F,
     &      X(198,I)*F,X(242,I)*F,
     &      X(243,I)*F,X(  7,I)*F,
     &      X( 11,I)*F
```

```
          END IF
   43   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,231) LABEL,SLDID
        WRITE(PR,144)
        DO 44 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6291) PHS(I),
     &      LOGFN(X(176,I)*F),LOGFN(X(177,I)*F),
     &      LOGFN(X(178,I)*F),LOGFN(X( 64,I)*F),
     &      LOGFN(X( 65,I)*F),LOGFN(X( 66,I)*F),
     &      LOGFN(X(281,I)*F),LOGFN(X(313,I)*F),
     &      LOGFN(TDCA(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6291) PHS(I),
     &      X(176,I)/T,X(177,I)/T,
     &      X(178,I)/T,X( 64,I)/T,
     &      X( 65,I)/T,X( 66,I)/T,
     &      X(281,I)/T,X(313,I)/T,
     &      TDCA(I)/T
          ELSE
            WRITE(PR,629) PHS(I),
     &      X(176,I)*F,X(177,I)*F,
     &      X(178,I)*F,X( 64,I)*F,
     &      X( 65,I)*F,X( 66,I)*F,
     &      X(281,I)*F,X(313,I)*F,
     &      TDCA(I)*F
          END IF
   44   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=190)
        WRITE(7,230) LABEL
        DO 86 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(  3,I)*F)
            Y(2 )=LOGFN(X( 13,I)*F)
            Y(3 )=LOGFN(X(  8,I)*F)
            Y(4 )=LOGFN(X(197,I)*F)
            Y(5 )=LOGFN(X(198,I)*F)
            Y(6 )=LOGFN(X(242,I)*F)
            Y(7 )=LOGFN(X(243,I)*F)
            Y(8 )=LOGFN(X(  7,I)*F)
            Y(9 )=LOGFN(X( 11,I)*F)
            Y(10)=LOGFN(X(176,I)*F)
            Y(11)=LOGFN(X(177,I)*F)
            Y(12)=LOGFN(X(178,I)*F)
            Y(13)=LOGFN(X( 64,I)*F)
            Y(14)=LOGFN(X( 65,I)*F)
            Y(15)=LOGFN(X( 66,I)*F)
            Y(16)=LOGFN(X(281,I)*F)
            Y(17)=LOGFN(X(281,I)*F)
            Y(18)=LOGFN(TDCA(I)*F)
            WRITE(7,6112) PHS(I),(Y(J),J=1,18)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6113) PHS(I),
     &      X(  3,I)/T,X( 13,I)/T,
     &      X(  8,I)/T,X(197,I)/T,
     &      X(198,I)/T,X(242,I)/T,
     &      X(243,I)/T,X(  7,I)/T,
     &      X( 11,I)/T,
     &      X(176,I)/T,X(177,I)/T,
     &      X(178,I)/T,X( 64,I)/T,
     &      X( 65,I)/T,X( 66,I)/T,
     &      X(281,I)/T,X(313,I)/T,
     &      TDCA(I)/T
          ELSE
            WRITE(7,6112) PHS(I),
     &      X(  3,I)*F,X( 13,I)*F,
     &      X(  8,I)*F,X(197,I)*F,
     &      X(198,I)*F,X(242,I)*F,
     &      X(243,I)*F,X(  7,I)*F,
     &      X( 11,I)*F,
     &      X(176,I)*F,X(177,I)*F,
     &      X(178,I)*F,X( 64,I)*F,
     &      X( 65,I)*F,X( 66,I)*F,
     &      X(281,I)*F,X(313,I)*F,
     &      TDCA(I)*F
          END IF
   86   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 230  FORMAT(     '      CALCIUM SYSTEM IN 'A38)
 231  FORMAT(/T44,'      CALCIUM SYSTEM IN ',A38//T50,'SOLID = ',A27)
 629  FORMAT(/,F5.2,1P2D13.4,2D15.4,5D13.4)
 6291 FORMAT(/,F5.2,  2F13.4,2F15.4,5F13.4)
 611  FORMAT(/,F5.2,1P6D13.4,D14.4,D15.4,D14.4,D12.4)
 6111 FORMAT(/,F5.2,  6F13.4,F14.4,F15.4,F14.4,F12.4)
 6112 FORMAT(F5.2,18F10.4)
 6113 FORMAT(F5.2,18D6.1)
 143  FORMAT(/'  pH      Ca(+2)       CaOH(+)    ',
     & ' CaSO4(0)    CaEDTA(-2)   CaHEDTA(-)    CaNTA(-)    ',
     & ' Ca(NTA)2(-4)      CaCO3(0)        CaHCO3(+)')
 144  FORMAT(/'  pH    CaSiO4(-2)   CaHSiO4(-)  ',
  404       & ' Ca(HSiO4)2(-4)    CaPO4(-)     CaHPO4(0)  ',
  405       & ' CaH2PO4(+)       SSCa        CaF(+)        T.D.Ca')
  406   C
  407         RETURN
  408         END


WCA  Local Symbols

Name                       Class   Type              Size   Offset

MTLFILE . . . . . . . . . . param                           0006
UNIT. . . . . . . . . . . . param                           000a
T1. . . . . . . . . . . . . param                           000e
TDCA. . . . . . . . . . . . param                           0012
I . . . . . . . . . . . . . local  INTEGER*4            4   0156
J . . . . . . . . . . . . . local  INTEGER*4            4   015a
T . . . . . . . . . . . . . local  REAL*8               8   015e
ISTART. . . . . . . . . . . local  INTEGER*4            4   0166
Y . . . . . . . . . . . . . local  REAL*8             144   016a
LABEL . . . . . . . . . . . local  CHAR*38             38   01fa
IEND. . . . . . . . . . . . local  INTEGER*4            4   0220
SKIP. . . . . . . . . . . . local  INTEGER*4            4   0224
PHS . . . . . . . . . . . . OUT    REAL*4             164   98b8
X . . . . . . . . . . . . . OUT    REAL*8          104632   0000
F . . . . . . . . . . . . . W      REAL*8               8   0000
F2. . . . . . . . . . . . . W      REAL*8               8   0008
F3. . . . . . . . . . . . . W      REAL*8               8   0010
NUMPH . . . . . . . . . . . W      INTEGER*4            4   0018
NPL . . . . . . . . . . . . W      INTEGER*4            4   001c
PR. . . . . . . . . . . . . W      INTEGER*4            4   003c
SLDID . . . . . . . . . . . W      CHAR*27             27   0020

  409   C
  410         SUBROUTINE WFEIII(TDFE3,T1,UNIT,MTLFILE)
  411   C     THIS BLOCK WRITES THE IRON(III) SUBSYSTEM AQUEOUS SPECIES REPORT.
  412   C
  413         REAL*4 PHS(41),TDFE3(41)
  414         REAL*8 X(319,41),Y(31)
  415         COMMON /OUT/X,PHS
  416         INTEGER I,J,ISTART,IEND,SKIP,UNIT
  417         REAL*8 F,F2,F3,T,T1
  418         INTEGER NUMPH,NPL,PR
  419         CHARACTER LABEL*38,SLDID*27
  420         CHARACTER MTLFILE*15
  421         REAL*8 LOGFN
  422         COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
  423   C
  424         ISTART=1
  425         IEND=0
  426         DO WHILE(IEND.NE.NUMPH)
```

```
      IEND=MIN0(IEND+NPL,NUMPH)
C
      CALL WHEAD
      IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES                 '
      IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES            '
      IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
      IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF IRON                       '
      IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF IRON                  '
      IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF IRON             '
      WRITE(PR,233) LABEL,SLDID
      WRITE(PR,145)
      DO 45 I=ISTART,IEND
        IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
          WRITE(PR,6201) PHS(I),
     &    LOGFN(X(184,I)*F)   ,LOGFN(X(152,I)*F),
     &    LOGFN(X(153,I)*F)   ,LOGFN(X(154,I)*F*F2),
     &    LOGFN(X(155,I)*F*F3),LOGFN(X(158,I)*F),
     &    LOGFN(X(159,I)*F)   ,LOGFN(X(199,I)*F),
     &    LOGFN(X(200,I)*F)
        ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
          T=1D0
          IF (T1.NE.0D0) T=T1/100
          WRITE(PR,6201) PHS(I),
     &    X(184,I)/T   ,X(152,I)/T,
     &    X(153,I)/T   ,X(154,I)/T*F2,
     &    X(155,I)/T*F3,X(158,I)/T,
     &    X(159,I)/T   ,X(199,I)/T,
     &    X(200,I)/T
        ELSE
          WRITE(PR,620) PHS(I),
     &    X(184,I)*F   ,X(152,I)*F,
     &    X(153,I)*F   ,X(154,I)*F*F2,
     &    X(155,I)*F*F3,X(158,I)*F,
     &    X(159,I)*F   ,X(199,I)*F,
     &    X(200,I)*F
        END IF
   45 CONTINUE
      SKIP=2*(NPL+ISTART-IEND)
      CALL WFOOT(SKIP)
C
      CALL WHEAD
      WRITE(PR,233) LABEL,SLDID
      WRITE(PR,146)
      DO 46 I=ISTART,IEND
        IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
          WRITE(PR,6211) PHS(I),
     &    LOGFN(X(201,I)*F)   ,LOGFN(X(202,I)*F),
     &    LOGFN(X(203,I)*F*F2),LOGFN(X(244,I)*F),
     &    LOGFN(X(245,I)*F)   ,LOGFN(X(246,I)*F),
     &    LOGFN(X(247,I)*F)   ,LOGFN(X(160,I)*F),
     &    LOGFN(X(161,I)*F)
```

```
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6211) PHS(I),
     &      X(201,I)/T   ,X(202,I)/T,
     &      X(203,I)/T*F2,X(244,I)/T,
     &      X(245,I)/T   ,X(246,I)/T,
     &      X(247,I)/T   ,X(160,I)/T,
     &      X(161,I)/T
          ELSE
            WRITE(PR,621) PHS(I),
     &      X(201,I)*F   ,X(202,I)*F,
     &      X(203,I)*F*F2,X(244,I)*F,
     &      X(245,I)*F   ,X(246,I)*F,
     &      X(247,I)*F   ,X(160,I)*F,
     &      X(161,I)*F
          END IF
   46   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,233) LABEL,SLDID
        WRITE(PR,147)
        DO 47 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6221) PHS(I),
     &      LOGFN(X(162,I)*F),LOGFN(X(166,I)*F),
     &      LOGFN(X(167,I)*F),LOGFN(X(156,I)*F),
     &      LOGFN(X(157,I)*F),LOGFN(X(163,I)*F),
     &      LOGFN(X(164,I)*F),LOGFN(X(165,I)*F),
     &      LOGFN(X(274,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6221) PHS(I),
     &      X(162,I)/T,X(166,I)/T,
     &      X(167,I)/T,X(156,I)/T,
     &      X(157,I)/T,X(163,I)/T,
     &      X(164,I)/T,X(165,I)/T,
     &      X(274,I)/T
          ELSE
            WRITE(PR,622) PHS(I),
     &      X(162,I)*F,X(166,I)*F,
     &      X(167,I)*F,X(156,I)*F,
     &      X(157,I)*F,X(163,I)*F,
     &      X(164,I)*F,X(165,I)*F,
     &      X(274,I)*F
          END IF
   47   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,233) LABEL,SLDID
        WRITE(PR,1471)
        DO I=ISTART,IEND
```

```
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6221) PHS(I),
     &      LOGFN(X(317,I)*F),LOGFN(X(318,I)*F),
     &      LOGFN(X(319,I)*F),LOGFN(TDFE3(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6221) PHS(I),
     &      X(317,I)/T,X(318,I)/T,
     &      X(319,I)/T,TDFE3(I)/T
          ELSE
            WRITE(PR,622) PHS(I),
     &      X(317,I)*F,X(318,I)*F,
     &      X(319,I)*F,TDFE3(I)*F
          END IF
        END DO
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &                RECL=320)
        WRITE(7,232) LABEL
        DO 87 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(184,I)*F)
            Y(2 )=LOGFN(X(152,I)*F)
            Y(3 )=LOGFN(X(153,I)*F)
            Y(4 )=LOGFN(X(154,I)*F*F2)
            Y(5 )=LOGFN(X(155,I)*F*F3)
            Y(6 )=LOGFN(X(158,I)*F)
            Y(7 )=LOGFN(X(159,I)*F)
            Y(8 )=LOGFN(X(199,I)*F)
            Y(9 )=LOGFN(X(200,I)*F)
            Y(10)=LOGFN(X(201,I)*F)
            Y(11)=LOGFN(X(202,I)*F)
            Y(12)=LOGFN(X(203,I)*F*F2)
            Y(13)=LOGFN(X(244,I)*F)
            Y(14)=LOGFN(X(245,I)*F)
            Y(15)=LOGFN(X(246,I)*F)
            Y(16)=LOGFN(X(247,I)*F)
            Y(17)=LOGFN(X(160,I)*F)
            Y(18)=LOGFN(X(161,I)*F)
            Y(19)=LOGFN(X(162,I)*F)
            Y(20)=LOGFN(X(166,I)*F)
            Y(21)=LOGFN(X(167,I)*F)
            Y(22)=LOGFN(X(156,I)*F)
            Y(23)=LOGFN(X(157,I)*F)
            Y(24)=LOGFN(X(163,I)*F)
            Y(25)=LOGFN(X(164,I)*F)
            Y(26)=LOGFN(X(165,I)*F)
            Y(27)=LOGFN(X(274,I)*F)
            Y(28)=LOGFN(X(317,I)*F)
            Y(29)=LOGFN(X(318,I)*F)
            Y(30)=LOGFN(X(319,I)*F)
            Y(31)=LOGFN(TDFE3(I)*F)
            WRITE(7,6222) PHS(I),(Y(J),J=1,31)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6223) PHS(I),
     &      X(184,I)/T   ,X(152,I)/T,
     &      X(153,I)/T   ,X(154,I)/T*F2,
     &      X(155,I)/T*F3,X(158,I)/T,
     &      X(159,I)/T   ,X(199,I)/T,
     &      X(200,I)/T,
     &      X(201,I)/T   ,X(202,I)/T,
     &      X(203,I)/T*F2,X(244,I)/T,
     &      X(245,I)/T   ,X(246,I)/T,
     &      X(247,I)/T   ,X(160,I)/T,
     &      X(161,I)/T,
     &      X(162,I)/T,X(166,I)/T,
     &      X(167,I)/T,X(156,I)/T,
     &      X(157,I)/T,X(163,I)/T,
     &      X(164,I)/T,X(165,I)/T,
     &      X(274,I)/T,TDFE3(I)/T
          ELSE
            WRITE(7,6222) PHS(I),
     &      X(184,I)*F   ,X(152,I)*F,
     &      X(153,I)*F   ,X(154,I)*F*F2,
     &      X(155,I)*F*F3,X(158,I)*F,
     &      X(159,I)*F   ,X(199,I)*F,
     &      X(200,I)*F,
     &      X(201,I)*F   ,X(202,I)*F,
     &      X(203,I)*F*F2,X(244,I)*F,
     &      X(245,I)*F   ,X(246,I)*F,
     &      X(247,I)*F   ,X(160,I)*F,
     &      X(161,I)*F,
     &      X(162,I)*F,X(166,I)*F,
     &      X(167,I)*F,X(156,I)*F,
     &      X(157,I)*F,X(163,I)*F,
     &      X(164,I)*F,X(165,I)*F,
     &      X(274,I)*F,TDFE3(I)*F
          END IF
   87   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 232  FORMAT(     '      IRON(III) SYSTEM IN ',A38)
 233  FORMAT(/T44,'      IRON(III) SYSTEM IN ',A38//T50,'SOLID = ',A27)
 620  FORMAT(/,F5.2,1P3D13.4,2D15.4,2(D14.4,D13.4))
 6201 FORMAT(/,F5.2,  3F13.4,2F15.4,2(F14.4,F13.4))
 621  FORMAT(/,F5.2,1PD13.4,D15.4,D17.4,D14.4,2D13.4,3D14.4)
 6211 FORMAT(/,F5.2,  F13.4,F15.4,F17.4,F14.4,2F13.4,3F14.4)
 622  FORMAT(/,F5.2,1PD12.4,5D13.4,4D12.4)
```

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  631   6221 FORMAT(/,F5.2,  F12.4,5F13.4,4F12.4)
  632   6222 FORMAT(F5.2,31F10.4)
  633   6223 FORMAT(F5.2,31F6.1)
  634   6224 FORMAT(/,F5.2,4D12.4)
  635   145  FORMAT(/'  pH      Fe(+3)      FeOH(+2)    ',
  636       & ' Fe(OH)2(+)    Fe2(OH)2(+4)   Fe3(OH)4(+5)    FeSO4(+)    ',
  637       & ' Fe(SO4)2(-)    FeEDTA(-)   FeHEDTA(0)')
  638   146  FORMAT(/'  pH   FeOHEDTA(-2)  Fe(OH)2EDTA(-3) ',
  639       & ' (FeOHEDTA)2(-4)  FeNTA(0)   Fe(NTA)2(-3) ',
  640       & ' FeOHNTA(-)  Fe(OH)2NTA(-2)   FeCl(+2)      FeCl2(+) ')
  641   147  FORMAT(/'  pH     FeCl3(0)   FeHSeO3(+2)  ',
  642       & ' FeHSi04(0)    FeHPO4(+)   FeH2PO4(+2)   FeAc(+2)    ',
  643       & ' FeAc2(+)     FeAc3(0)    SSFe(III)')
  644   1471 FORMAT(/'  pH     FeF(+2)        FeF2(-)      FeF3(0)    ',
  645       & ' T.D.Fe(III)')
  646   C
  647         RETURN
  648         END


WFEIII  Local Symbols

Name                        Class   Type              Size   Offset

MTLFILE . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . param                             000e
TDFE3 . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . local   INTEGER*4             4   0228
J . . . . . . . . . . . . . local   INTEGER*4             4   022c
T . . . . . . . . . . . . . local   REAL*8                8   0230
ISTART. . . . . . . . . . . local   INTEGER*4             4   0238
Y . . . . . . . . . . . . . local   REAL*8              248   023c
LABEL . . . . . . . . . . . local   CHAR*38              38   0334
IEND. . . . . . . . . . . . local   INTEGER*4             4   035a
SKIP. . . . . . . . . . . . local   INTEGER*4             4   035e
PHS . . . . . . . . . . . . OUT     REAL*4              164   98b8
X . . . . . . . . . . . . . OUT     REAL*8           104632   0000
F . . . . . . . . . . . . . W       REAL*8                8   0000
F2. . . . . . . . . . . . . W       REAL*8                8   0008
F3. . . . . . . . . . . . . W       REAL*8                8   0010
NUMPH . . . . . . . . . . . W       INTEGER*4             4   0018
NPL . . . . . . . . . . . . W       INTEGER*4             4   001c
PR. . . . . . . . . . . . . W       INTEGER*4             4   003c
SLDID . . . . . . . . . . . W       CHAR*27              27   0020

  649   C
  650         SUBROUTINE WHGI(TDHG1,T1,UNIT,MTLFILE)
  651   C     THIS BLOCK WRITES THE MERCURY(I) SUBSYSTEM AQUEOUS SPECIES REPORT
  652   C
  653         REAL*4 PHS(41),TDHG1(41)
```

```
      REAL*8 X(319,41),Y(6)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF HG2(++)              '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF HG2(++)         '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF HG2(++)    '
        WRITE(PR,235) LABEL,SLDID
        WRITE(PR,148)
        DO 48 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6271) PHS(I),
     &      LOGFN(X(183,I)*F),LOGFN(X(149,I)*F),
     &      LOGFN(X(150,I)*F),LOGFN(X(151,I)*F),
     &      LOGFN(X(282,I)*F),LOGFN(TDHG1(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/1D2
            WRITE(PR,6271) PHS(I),
     &      X(183,I)/T,X(149,I)/T,
     &      X(150,I)/T,X(151,I)/T,
     &      X(282,I)/T,TDHG1(I)/T
          ELSE
            WRITE(PR,627) PHS(I),
     &      X(183,I)*F,X(149,I)*F,
     &      X(150,I)*F,X(151,I)*F,
     &      X(282,I)*F,TDHG1(I)*F
          END IF
   48   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=70)
        WRITE(7,234) LABEL
        DO 88 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(183,I)*F)
            Y(2 )=LOGFN(X(149,I)*F)
            Y(3 )=LOGFN(X(150,I)*F)
                Y(4 )=LOGFN(X(151,I)*F)
            Y(5 )=LOGFN(X(282,I)*F)
                Y(6 )=LOGFN(TDHG1(I)*F)
            WRITE(7,6272) PHS(I),(Y(J),J=1,6)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6273) PHS(I),
     &      X(183,I)/T,X(149,I)/T,
     &      X(150,I)/T,X(151,I)/T,
     &      X(282,I)/T,TDHG1(I)/T
          ELSE
            WRITE(7,6272) PHS(I),
     &      X(183,I)*F,X(149,I)*F,
     &      X(150,I)*F,X(151,I)*F,
     &      X(282,I)*F,TDHG1(I)*F
          END IF
   88   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 234  FORMAT(     '   MERCURY(I) SYSTEM IN ',A38)
 235  FORMAT(/T44,'   MERCURY(I) SYSTEM IN ',A38//T50,'SOLID = ',A27)
 627  FORMAT(/,F5.2,1P4D14.4,D15.4,D14.4)
 6271 FORMAT(/,F5.2,  4F14.4,F15.4,F14.4)
 6272 FORMAT(F5.2,6F10.4)
 6273 FORMAT(F5.2,6F6.1)
 148  FORMAT(/'  pH        Hg2(+2)      Hg2OH(+)      ',
     & ' Hg2SO4(0)   Hg2(SO4)2(-2)      SSHg(I)      T.D.Hg(I)')
C
      RETURN
      END
```

```
WHGI  Local Symbols

Name                        Class   Type               Size   Offset

MTLFILE . . . . . . . . . . param                              0006
UNIT. . . . . . . . . . . . param                              000a
T1. . . . . . . . . . . . . param                              000e
TDHG1 . . . . . . . . . . . param                              0012
I . . . . . . . . . . . . . local   INTEGER*4             4    0362
J . . . . . . . . . . . . . local   INTEGER*4             4    0366
T . . . . . . . . . . . . . local   REAL*8                8    036a
ISTART. . . . . . . . . . . local   INTEGER*4             4    0372
```

```
WHGI  Local Symbols

Name                          Class   Type              Size   Offset

Y . . . . . . . . . . . . . . local   REAL*8              48   0376
LABEL . . . . . . . . . . . . local   CHAR*38             38   03a6
IEND. . . . . . . . . . . . . local   INTEGER*4            4   03cc
SKIP. . . . . . . . . . . . . local   INTEGER*4            4   03d0
PHS . . . . . . . . . . . . . OUT     REAL*4             164   98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632   0000
F . . . . . . . . . . . . . . W       REAL*8               8   0000
F2. . . . . . . . . . . . . . W       REAL*8               8   0008
F3. . . . . . . . . . . . . . W       REAL*8               8   0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4   0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4   001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4   003c
SLDID . . . . . . . . . . . . W       CHAR*27             27   0020

Global Symbols

Name                          Class   Type              Size   Offset

LOGFN . . . . . . . . . . . . extern  REAL*8             ***      ***
OUT . . . . . . . . . . . . . common  ***             104796     0000
W . . . . . . . . . . . . . . common  ***                 64     0000
WCA . . . . . . . . . . . . . FSUBRT  ***                ***     3bdb
WFEIII. . . . . . . . . . . . FSUBRT  ***                ***     648b
WFOOT . . . . . . . . . . . . extern  ***                ***      ***
WHEAD . . . . . . . . . . . . extern  ***                ***      ***
WHGI. . . . . . . . . . . . . FSUBRT  ***                ***     a884
WMG . . . . . . . . . . . . . FSUBRT  ***                ***     1532
WNA . . . . . . . . . . . . . FSUBRT  ***                ***     0000

Code size = b73a (46906)
Data size = 099b (2459)
Bss size  = 03d4 (980)

No errors detected
```

```
      SUBROUTINE WHGII(TDHG2,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE MERCURY(II) SUBSYSTEM AQUEOUS SPECIES REPOR
C
      REAL*4 PHS(41),TDHG2(41)
      REAL*8 X(319,41),Y(28)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES             '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES        '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF MERCURY                '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF MERCURY           '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF MERCURY      '
        WRITE(PR,237) LABEL,SLDID
        WRITE(PR,149)
        DO 49 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6411) PHS(I),
     &      LOGFN(X(103,I)*F),LOGFN(X( 84,I)*F),
     &      LOGFN(X( 85,I)*F),LOGFN(X( 86,I)*F),
     &      LOGFN(X( 87,I)*F*F2),LOGFN(X( 88,I)*F*F3),
     &      LOGFN(X(100,I)*F),LOGFN(X(101,I)*F),
     &      LOGFN(X(204,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6411) PHS(I),
     &      X(103,I)/T,X( 84,I)/T,
     &      X( 85,I)/T,X( 86,I)/T,
     &      X( 87,I)/T*F2,X( 88,I)/T*F3,
     &      X(100,I)/T,X(101,I)/T,
     &      X(204,I)/T
          ELSE
            WRITE(PR,641) PHS(I),
     &      X(103,I)*F,X( 84,I)*F,
     &      X( 85,I)*F,X( 86,I)*F,
     &      X( 87,I)*F*F2,X( 88,I)*F*F3,
     &      X(100,I)*F,X(101,I)*F,
     &       X(204,I)*F
          END IF
   49   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,237) LABEL,SLDID
        WRITE(PR,150)
        DO 50 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6231) PHS(I),
     &      LOGFN(X(205,I)*F),LOGFN(X(206,I)*F),
     &      LOGFN(X(248,I)*F),LOGFN(X( 90,I)*F),
     &      LOGFN(X( 91,I)*F),LOGFN(X( 92,I)*F),
     &      LOGFN(X( 89,I)*F),LOGFN(X(102,I)*F),
     &      LOGFN(X(109,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6231) PHS(I),
     &      X(205,I)/T,X(206,I)/T,
     &      X(248,I)/T,X( 90,I)/T,
     &      X( 91,I)/T,X( 92,I)/T,
     &      X( 89,I)/T,X(102,I)/T,
     &      X(109,I)/T
          ELSE
            WRITE(PR,623) PHS(I),
     &      X(205,I)*F,X(206,I)*F,
     &      X(248,I)*F,X( 90,I)*F,
     &      X( 91,I)*F,X( 92,I)*F,
     &      X( 89,I)*F,X(102,I)*F,
     &      X(109,I)*F
          END IF
   50   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,237) LABEL,SLDID
        WRITE(PR,151)
        DO 99 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6241) PHS(I),
     &      LOGFN(X( 93,I)*F),LOGFN(X( 94,I)*F),
     &      LOGFN(X( 95,I)*F),LOGFN(X( 96,I)*F),
     &      LOGFN(X( 97,I)*F),LOGFN(X( 98,I)*F),
     &      LOGFN(X( 99,I)*F),LOGFN(X(283,I)*F),
     &      LOGFN(X(315,I)*F),LOGFN(TDHG2(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6241) PHS(I),
     &      X( 93,I)/T,X( 94,I)/T,
     &      X( 95,I)/T,X( 96,I)/T,
     &      X( 97,I)/T,X( 98,I)/T,
     &      X( 99,I)/T,X(283,I)/T,
     &      X(315,I)/T,TDHG2(I)/T
          ELSE
            WRITE(PR,624) PHS(I),
     &      X( 93,I)*F,X( 94,I)*F,
     &      X( 95,I)*F,X( 96,I)*F,
     &      X( 97,I)*F,X( 98,I)*F,
     &      X( 99,I)*F,X(283,I)*F,
     &      X(315,I)*F,TDHG2(I)*F
          END IF
   99   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &                RECL=290)
        WRITE(7,236) LABEL
        DO 89 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(103,I)*F)
            Y(2 )=LOGFN(X( 84,I)*F)
            Y(3 )=LOGFN(X( 85,I)*F)
            Y(4 )=LOGFN(X( 86,I)*F)
            Y(5 )=LOGFN(X( 87,I)*F*F2)
            Y(6 )=LOGFN(X( 88,I)*F*F3)
            Y(7 )=LOGFN(X(100,I)*F)
            Y(8 )=LOGFN(X(101,I)*F)
            Y(9 )=LOGFN(X(204,I)*F)
            Y(10)=LOGFN(X(205,I)*F)
            Y(11)=LOGFN(X(206,I)*F)
            Y(12)=LOGFN(X(248,I)*F)
            Y(13)=LOGFN(X( 90,I)*F)
            Y(14)=LOGFN(X( 91,I)*F)
            Y(15)=LOGFN(X( 92,I)*F)
            Y(16)=LOGFN(X( 89,I)*F)
            Y(17)=LOGFN(X(102,I)*F)
            Y(18)=LOGFN(X(109,I)*F)
            Y(19)=LOGFN(X( 93,I)*F)
            Y(20)=LOGFN(X( 94,I)*F)
            Y(21)=LOGFN(X( 95,I)*F)
            Y(22)=LOGFN(X( 96,I)*F)
            Y(23)=LOGFN(X( 97,I)*F)
            Y(24)=LOGFN(X( 98,I)*F)
            Y(25)=LOGFN(X( 99,I)*F)
            Y(26)=LOGFN(X(283,I)*F)
            Y(27)=LOGFN(X(315,I)*F)
            Y(28)=LOGFN(TDHG2(I)*F)
            WRITE(7,6242) PHS(I),(Y(J),J=1,28)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
```

```
            WRITE(7,6243) PHS(I),
     &      X(103,I)/T,X( 84,I)/T,
     &      X( 85,I)/T,X( 86,I)/T,
     &      X( 87,I)/T*F2,X( 88,I)/T*F3,
     &      X(100,I)/T,X(101,I)/T,
     &      X(204,I)/T,
     &      X(205,I)/T,X(206,I)/T,
     &      X(248,I)/T,X( 90,I)/T,
     &      X( 91,I)/T,X( 92,I)/T,
     &      X( 89,I)/T,X(102,I)/T,
     &      X(109,I)/T,
     &      X( 93,I)/T,X( 94,I)/T,
     &      X( 95,I)/T,X( 96,I)/T,
     &      X( 97,I)/T,X( 98,I)/T,
     &      X( 99,I)/T,X(283,I)/T,
     &      TDHG2(I)/T
          ELSE
            WRITE(7,6242) PHS(I),
     &      X(103,I)*F,X( 84,I)*F,
     &      X( 85,I)*F,X( 86,I)*F,
     &      X( 87,I)*F*F2,X( 88,I)*F*F3,
     &      X(100,I)*F,X(101,I)*F,
     &      X(204,I)*F,
     &      X(205,I)*F,X(206,I)*F,
     &      X(248,I)*F,X( 90,I)*F,
     &      X( 91,I)*F,X( 92,I)*F,
     &      X( 89,I)*F,X(102,I)*F,
     &      X(109,I)*F,
     &      X( 93,I)*F,X( 94,I)*F,
     &      X( 95,I)*F,X( 96,I)*F,
     &      X( 97,I)*F,X( 98,I)*F,
     &      X( 99,I)*F,X(283,I)*F,
     &      TDHG2(I)*F
          END IF
   89   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 236  FORMAT(     '  MERCURY(II) SYSTEM IN ',A38)
 237  FORMAT(/T44,'  MERCURY(II) SYSTEM IN ',A38//T50,'SOLID = ',A27)
 641  FORMAT(/,F5.2,1P9D14.4)
 6411 FORMAT(/,F5.2,  9F14.4)
 623  FORMAT(/,F5.2,1PD13.4,8D14.4)
 6231 FORMAT(/,F5.2,  F13.4,8F14.4)
 624  FORMAT(/,F5.2,1PD13.4,D14.4,D15.4,7D14.4)
 6241 FORMAT(/,F5.2,  F13.4,F14.4,F15.4,7F14.4)
 6242 FORMAT(F5.2,28F10.4)
 6243 FORMAT(F5.2,28F6.1)
 149  FORMAT(/' pH        Hg(+2)        HgOH(+)    ',
     & ' Hg(OH)2(0)    Hg(OH)3(-)     Hg2OH(+3)   Hg3(OH)3(+3)   ',
     & ' HgSO4(0)    Hg(SO4)2(-2)   HgEDTA(-2)')
 Line#  Source Line         Microsoft FORTRAN Optimizing Compiler Version 5.00

   205    150  FORMAT(/'  pH    HgHEDTA(-)   HgOHEDTA(-3)    ',
   206        & ' HgNTA(-)        HgCl(+)      HgCl2(0)       HgCl3(-)      ',
   207        & ' HgOHCl(0)       HgA2(0)    Hg(SeO3)2(-2)')
   208    151  FORMAT(/'  pH      HgNH3(+2)   Hg(NH3)2(+2)   ',
   209        & ' Hg(NH3)3(+2)      HgAc(+)       HgAc2(0)       HgAc3(-)     ',
   210        & ' Hg(HS)2(0)      SSHg(II)        HgF(+)      T.D.Hg(II)')
   211  C
   212          RETURN
   213          END


WHGII  Local Symbols

Name                           Class   Type                Size    Offset

MTLFILE . . . . . . . . . . . param                                 0006
UNIT. . . . . . . . . . . . . param                                 000a
T1. . . . . . . . . . . . . . param                                 000e
TDHG2 . . . . . . . . . . . . param                                 0012
I . . . . . . . . . . . . . . local   INTEGER*4              4      0002
J . . . . . . . . . . . . . . local   INTEGER*4              4      0006
T . . . . . . . . . . . . . . local   REAL*8                 8      000a
ISTART. . . . . . . . . . . . local   INTEGER*4              4      0012
Y . . . . . . . . . . . . . . local   REAL*8               224      0016
LABEL . . . . . . . . . . . . local   CHAR*38               38      00f6
IEND. . . . . . . . . . . . . local   INTEGER*4              4      011c
SKIP. . . . . . . . . . . . . local   INTEGER*4              4      0120
PHS . . . . . . . . . . . . . OUT     REAL*4               164      98b8
X . . . . . . . . . . . . . . OUT     REAL*8            104632      0000
F . . . . . . . . . . . . . . W       REAL*8                 8      0000
F2. . . . . . . . . . . . . . W       REAL*8                 8      0008
F3. . . . . . . . . . . . . . W       REAL*8                 8      0010
NUMPH . . . . . . . . . . . . W       INTEGER*4              4      0018
NPL . . . . . . . . . . . . . W       INTEGER*4              4      001c
PR. . . . . . . . . . . . . . W       INTEGER*4              4      003c
SLDID . . . . . . . . . . . . W       CHAR*27               27      0020

   214  C
   215          SUBROUTINE WCD(TDCD,T1,UNIT,MTLFILE)
   216  C       THIS BLOCK WRITES THE CADMIUM SUBSYSTEM AQUEOUS SPECIES REPORT.
   217  C
   218          REAL*4 PHS(41),TDCD(41)
   219          REAL*8 X(319,41),Y(28)
   220          COMMON /OUT/X,PHS
   221          INTEGER I,J,ISTART,IEND,SKIP,UNIT
   222          REAL*8 F,F2,F3,T,T1
   223          INTEGER NUMPH,NPL,PR
   224          CHARACTER LABEL*38,SLDID*27
   225          CHARACTER MTLFILE*15
   226          REAL*8 LOGFN
   227          COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
```

```
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF CADMIUM              '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF CADMIUM         '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF CADMIUM    '
        WRITE(PR,239) LABEL,SLDID
        WRITE(PR,152)
        DO 52 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6371) PHS(I),
     &      LOGFN(X( 40,I)*F),LOGFN(X( 41,I)*F),
     &      LOGFN(X( 42,I)*F),LOGFN(X( 43,I)*F),
     &      LOGFN(X( 44,I)*F),LOGFN(X( 45,I)*F*F2),
     &      LOGFN(X( 46,I)*F),LOGFN(X(207,I)*F),
     &      LOGFN(X(208,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6371) PHS(I),
     &      X( 40,I)/T,X( 41,I)/T,
     &      X( 42,I)/T,X( 43,I)/T,
     &      X( 44,I)/T,X( 45,I)/T*F2,
     &      X( 46,I)/T,X(207,I)/T,
     &      X(208,I)/T
          ELSE
            WRITE(PR,637) PHS(I),
     &      X( 40,I)*F,X( 41,I)*F,
     &      X( 42,I)*F,X( 43,I)*F,
     &      X( 44,I)*F,X( 45,I)*F*F2,
     &      X( 46,I)*F,X(207,I)*F,
     &      X(208,I)*F
          END IF
   52   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,239) LABEL,SLDID
        WRITE(PR,153)
        DO 53 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6251) PHS(I),
     &      LOGFN(X(249,I)*F),LOGFN(X(250,I)*F),
     &      LOGFN(X(251,I)*F),LOGFN(X( 47,I)*F),
     &      LOGFN(X( 48,I)*F),LOGFN(X( 49,I)*F),
     &      LOGFN(X( 63,I)*F),LOGFN(X(110,I)*F),
     &      LOGFN(X( 50,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6251) PHS(I),
     &      X(249,I)/T,X(250,I)/T,
     &      X(251,I)/T,X( 47,I)/T,
     &      X( 48,I)/T,X( 49,I)/T,
     &      X( 63,I)/T,X(110,I)/T,
     &      X( 50,I)/T
          ELSE
            WRITE(PR,625) PHS(I),
     &      X(249,I)*F,X(250,I)*F,
     &      X(251,I)*F,X( 47,I)*F,
     &      X( 48,I)*F,X( 49,I)*F,
     &      X( 63,I)*F,X(110,I)*F,
     &      X( 50,I)*F
          END IF
   53   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,239) LABEL,SLDID
        WRITE(PR,154)
        DO 54 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6261) PHS(I),
     &      LOGFN(X( 51,I)*F)   ,LOGFN(X( 52,I)*F),
     &      LOGFN(X( 56,I)*F)   ,LOGFN(X( 57,I)*F),
     &      LOGFN(X( 58,I)*F)   ,LOGFN(X( 81,I)*F),
     &      LOGFN(X( 82,I)*F)   ,LOGFN(X( 83,I)*F),
     &      LOGFN(X(271,I)*F),LOGFN(TDCD(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6261) PHS(I),
     &      X( 51,I)/T,X( 52,I)/T,
     &      X( 56,I)/T,X( 57,I)/T,
     &      X( 58,I)/T,X( 81,I)/T,
     &      X( 82,I)/T,X( 83,I)/T,
     &      X(271,I)/T,TDCD(I)/T
          ELSE
            WRITE(PR,626) PHS(I),
     &      X( 51,I)*F,X( 52,I)*F,
     &      X( 56,I)*F,X( 57,I)*F,
     &      X( 58,I)*F,X( 81,I)*F,
     &      X( 82,I)*F,X( 83,I)*F,
     &      X(271,I)*F,TDCD(I)*F
          END IF
   54   CONTINUE
        CALL WFOOT(SKIP)
        ISTART=IEND+1
```

```
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=290)
        WRITE(7,238) LABEL
        DO 90 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X( 40,I)*F)
            Y(2 )=LOGFN(X( 41,I)*F)
            Y(3 )=LOGFN(X( 42,I)*F)
            Y(4 )=LOGFN(X( 43,I)*F)
            Y(5 )=LOGFN(X( 44,I)*F)
            Y(6 )=LOGFN(X( 45,I)*F*F2)
            Y(7 )=LOGFN(X( 46,I)*F)
            Y(8 )=LOGFN(X(207,I)*F)
            Y(9 )=LOGFN(X(208,I)*F)
            Y(10)=LOGFN(X(249,I)*F)
            Y(11)=LOGFN(X(250,I)*F)
            Y(12)=LOGFN(X(251,I)*F)
            Y(13)=LOGFN(X( 47,I)*F)
            Y(14)=LOGFN(X( 48,I)*F)
            Y(15)=LOGFN(X( 49,I)*F)
            Y(16)=LOGFN(X( 63,I)*F)
            Y(17)=LOGFN(X(110,I)*F)
            Y(18)=LOGFN(X( 50,I)*F)
            Y(19)=LOGFN(X( 51,I)*F)
            Y(20)=LOGFN(X( 52,I)*F)
            Y(21)=LOGFN(X( 56,I)*F)
            Y(22)=LOGFN(X( 57,I)*F)
            Y(23)=LOGFN(X( 58,I)*F)
            Y(24)=LOGFN(X( 81,I)*F)
            Y(25)=LOGFN(X( 82,I)*F)
            Y(26)=LOGFN(X( 83,I)*F)
            Y(27)=LOGFN(X(271,I)*F)
            Y(28)=LOGFN(TDCD(I)*F)
            WRITE(7,6262) PHS(I),(Y(J),J=1,28)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6263) PHS(I),
     &      X( 40,I)/T,X( 41,I)/T,
     &      X( 42,I)/T,X( 43,I)/T,
     &      X( 44,I)/T,X( 45,I)/T*F2,
     &      X( 46,I)/T,X(207,I)/T,
     &      X(208,I)/T,
     &      X(249,I)/T,X(250,I)/T,
     &      X(251,I)/T,X( 47,I)/T,
     &      X( 48,I)/T,X( 49,I)/T,
     &      X( 63,I)/T,X(110,I)/T,
     &      X( 50,I)/T,
     &      X( 51,I)/T,X( 52,I)/T,
     &      X( 56,I)/T,X( 57,I)/T,
     &      X( 58,I)/T,X( 81,I)/T,
     &      X( 82,I)/T,X( 83,I)/T,
     &      X(271,I)/T,TDCD(I)/T
          ELSE
            WRITE(7,6262) PHS(I),
     &      X( 40,I)*F,X( 41,I)*F,
     &      X( 42,I)*F,X( 43,I)*F,
     &      X( 44,I)*F,X( 45,I)*F*F2,
     &      X( 46,I)*F,X(207,I)*F,
     &      X(208,I)*F,
     &      X(249,I)*F,X(250,I)*F,
     &      X(251,I)*F,X( 47,I)*F,
     &      X( 48,I)*F,X( 49,I)*F,
     &      X( 63,I)*F,X(110,I)*F,
     &      X( 50,I)*F,
     &      X( 51,I)*F,X( 52,I)*F,
     &      X( 56,I)*F,X( 57,I)*F,
     &      X( 58,I)*F,X( 81,I)*F,
     &      X( 82,I)*F,X( 83,I)*F,
     &      X(271,I)*F,TDCD(I)*F
          END IF
   90   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 238  FORMAT(    '     CADMIUM SYSTEM IN ',A38)
 239  FORMAT(/T44,'    CADMIUM SYSTEM IN ',A38//T50,'SOLID = ',A27)
 637  FORMAT(/,F5.2,1P9D14.4)
 6371 FORMAT(/,F5.2,  9F14.4)
 625  FORMAT(/,F5.2,1P8D14.4,D15.4)
 6251 FORMAT(/,F5.2,  8F14.4,F15.4)
 626  FORMAT(/,F5.2,1PD13.4,D14.4,4D12.4,4D13.4)
 6261 FORMAT(/,F5.2,   F13.4,F14.4,4F12.4,4F13.4)
 6262 FORMAT(F5.2,28F10.4)
 6263 FORMAT(F5.2,28F6.1)
 152  FORMAT(/'  pH       Cd(+2)        CdOH(+)     ',
     & ' Cd(OH)2(0)    Cd(OH)3(-)    Cd(OH)4(-2)    Cd2OH(+3)    ',
     & ' CdSO4(0)     CdEDTA(-2)    CdHEDTA(-)')
 153  FORMAT(/'  pH      CdNTA(-)    Cd(NTA)2(-4)   ',
     & ' CdOHNTA(-2)     CdCl(+)      CdCl2(0)      ',
     & ' CdCl3(-)       CdA2(0)    Cd(SeO3)2(-2)    CdNH3(+2)')
 154  FORMAT(/'  pH   Cd(NH3)2(+2)  Cd(NH3)3(+2)    ',
     & ' CdAc(+)    CdAc2(0)    CdAc3(-)     CdHS(+)    ',
     & ' Cd(HS)2(0)   Cd(HS)3(-)       SSCd        T.D.Cd')
C
      RETURN
      END
```

```
WCD  Local Symbols

Name                          Class   Type              Size   Offset

MTLFILE . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . . param                             000e
TDCD. . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local   INTEGER*4            4    0124
J . . . . . . . . . . . . . . local   INTEGER*4            4    0128
T . . . . . . . . . . . . . . local   REAL*8               8    012c
ISTART. . . . . . . . . . . . local   INTEGER*4            4    0134
Y . . . . . . . . . . . . . . local   REAL*8             224    0138
LABEL . . . . . . . . . . . . local   CHAR*38             38    0218
IEND. . . . . . . . . . . . . local   INTEGER*4            4    023e
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    0242
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020
```

```
C
      SUBROUTINE WCU(TDCU,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE COPPER SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDCU(41)
      REAL*8 X(319,41),Y(26)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES              '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES         '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF COPPER                  '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF COPPER          '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF COPPER     '
        WRITE(PR,241) LABEL,SLDID
        WRITE(PR,155)
        DO 55 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6011) PHS(I),
     &      LOGFN(X( 18,I)*F),LOGFN(X( 21,I)*F),
     &      LOGFN(X( 22,I)*F),LOGFN(X( 23,I)*F),
     &      LOGFN(X( 24,I)*F),LOGFN(X( 25,I)*F*F2),
     &      LOGFN(X( 26,I)*F),LOGFN(X(209,I)*F),
     &      LOGFN(X(210,I)*F),LOGFN(X(211,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6011) PHS(I),
     &      X( 18,I)/T,X( 21,I)/T,
     &      X( 22,I)/T,X( 23,I)/T,
     &      X( 24,I)/T,X( 25,I)/T*F2,
     &      X( 26,I)/T,X(209,I)/T,
     &      X(210,I)/T,X(211,I)/T
          ELSE
            WRITE(PR,601) PHS(I),
     &      X( 18,I)*F,X( 21,I)*F,
     &      X( 22,I)*F,X( 23,I)*F,
     &      X( 24,I)*F,X( 25,I)*F*F2,
     &      X( 26,I)*F,X(209,I)*F,
     &      X(210,I)*F,X(211,I)*F
          END IF
   55   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,241) LABEL,SLDID
        WRITE(PR,156)
        DO 56 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6381) PHS(I),
     &      LOGFN(X(252,I)*F),LOGFN(X(253,I)*F),
     &      LOGFN(X(254,I)*F),LOGFN(X( 19,I)*F),
     &      LOGFN(X( 20,I)*F),LOGFN(X( 27,I)*F),
     &      LOGFN(X( 28,I)*F),LOGFN(X( 29,I)*F),
     &      LOGFN(X( 30,I)*F),LOGFN(X( 31,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6381) PHS(I),
     &      X(252,I)/T,X(253,I)/T,
     &      X(254,I)/T,X( 19,I)/T,
     &      X( 20,I)/T,X( 27,I)/T,
     &      X( 28,I)/T,X( 29,I)/T,
     &      X( 30,I)/T,X( 31,I)/T
```

```
          ELSE
            WRITE(PR,638) PHS(I),
     &      X(252,I)*F,X(253,I)*F,
     &      X(254,I)*F,X( 19,I)*F,
     &      X( 20,I)*F,X( 27,I)*F,
     &      X( 28,I)*F,X( 29,I)*F,
     &      X( 30,I)*F,X( 31,I)*F
          END IF
   56   CONTINUE
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,241) LABEL,SLDID
        WRITE(PR,157)
        DO 57 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6031) PHS(I),
     &      LOGFN(X( 32,I)*F),LOGFN(X( 33,I)*F),
     &      LOGFN(X( 36,I)*F),LOGFN(X( 37,I)*F),
     &      LOGFN(X(272,I)*F),LOGFN(X(314,I)*F),
     &      LOGFN(TDCU(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6031) PHS(I),
     &      X( 32,I)/T,X( 33,I)/T,
     &      X( 36,I)/T,X( 37,I)/T,
     &      X(272,I)/T,X(314,I)/T,
     &      TDCU(I)/T
          ELSE
            WRITE(PR,603) PHS(I),
     &      X( 32,I)*F,X( 33,I)*F,
     &      X( 36,I)*F,X( 37,I)*F,
     &      X(272,I)*F,X(314,I)*F,
     &      TDCU(I)*F
          END IF
   57   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=270)
        WRITE(7,240) LABEL
        DO 91 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X( 18,I)*F)
            Y(2 )=LOGFN(X( 21,I)*F)
            Y(3 )=LOGFN(X( 22,I)*F)
            Y(4 )=LOGFN(X( 23,I)*F)
            Y(5 )=LOGFN(X( 24,I)*F)
            Y(6 )=LOGFN(X( 25,I)*F*F2)
   554                   Y(7 )=LOGFN(X( 26,I)*F)
   555                   Y(8 )=LOGFN(X(209,I)*F)
   556                   Y(9 )=LOGFN(X(210,I)*F)
   557                   Y(10)=LOGFN(X(211,I)*F)
   558                   Y(11)=LOGFN(X(252,I)*F)
   559                   Y(12)=LOGFN(X(253,I)*F)
   560                   Y(13)=LOGFN(X(254,I)*F)
   561                   Y(14)=LOGFN(X( 19,I)*F)
   562                   Y(15)=LOGFN(X( 20,I)*F)
   563                   Y(16)=LOGFN(X( 27,I)*F)
   564                   Y(17)=LOGFN(X( 28,I)*F)
   565                   Y(18)=LOGFN(X( 29,I)*F)
   566                   Y(19)=LOGFN(X( 30,I)*F)
   567                   Y(20)=LOGFN(X( 31,I)*F)
   568                   Y(21)=LOGFN(X( 32,I)*F)
   569                   Y(22)=LOGFN(X( 33,I)*F)
   570                   Y(23)=LOGFN(X( 36,I)*F)
   571                   Y(24)=LOGFN(X( 37,I)*F)
   572                   Y(25)=LOGFN(X(272,I)*F)
   573                   Y(26)=LOGFN(X(314,I)*F)
   574                   Y(27)=LOGFN(TDCU(I)*F)
   575                   WRITE(7,6032) PHS(I),(Y(J),J=1,27)
   576                 ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
   577                   WRITE(7,6033) PHS(I),
   578         &         X( 18,I)/T,X( 21,I)/T,
   579         &         X( 22,I)/T,X( 23,I)/T,
   580         &         X( 24,I)/T,X( 25,I)/T*F2,
   581         &         X( 26,I)/T,X(209,I)/T,
   582         &         X(210,I)/T,X(211,I)/T,
   583         &         X(252,I)/T,X(253,I)/T,
   584         &         X(254,I)/T,X( 19,I)/T,
   585         &         X( 20,I)/T,X( 27,I)/T,
   586         &         X( 28,I)/T,X( 29,I)/T,
   587         &         X( 30,I)/T,X( 31,I)/T,
   588         &         X( 32,I)/T,X( 33,I)/T,
   589         &         X( 36,I)/T,X( 37,I)/T,
   590         &         X(272,I)/T,X(314,I)/T,
   591         &         TDCU(I)/T
   592                 ELSE
   593                   WRITE(7,6032) PHS(I),
   594         &         X( 18,I)*F,X( 21,I)*F,
   595         &         X( 22,I)*F,X( 23,I)*F,
   596         &         X( 24,I)*F,X( 25,I)*F*F2,
   597         &         X( 26,I)*F,X(209,I)*F,
   598         &         X(210,I)*F,X(211,I)*F,
   599         &         X(252,I)*F,X(253,I)*F,
   600         &         X(254,I)*F,X( 19,I)*F,
   601         &         X( 20,I)*F,X( 27,I)*F,
   602         &         X( 28,I)*F,X( 29,I)*F,
   603         &         X( 30,I)*F,X( 31,I)*F,
   604         &         X( 32,I)*F,X( 33,I)*F,
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  605       &      X( 36,I)*F,X( 37,I)*F,
  606       &      X(272,I)*F,X(314,I)*F,
  607       &      TDCU(I)*F
  608             END IF
  609     91    CONTINUE
  610           CLOSE (UNIT=7)
  611         END IF
  612  C
  613   240  FORMAT(     '     COPPER SYSTEM IN ',A38)
  614   241  FORMAT(/T44,'     COPPER SYSTEM IN ',A38//T50,'SOLID = ',A27)
  615   601  FORMAT(/,F5.2,1P5D12.4,D14.4,4D13.4)
  616   6011 FORMAT(/,F5.2,  5F12.4,F14.4,4F13.4)
  617   638  FORMAT(/,F5.2,1P6D13.4,4D12.4)
  618   6381 FORMAT(/,F5.2,  6F13.4,4F12.4)
  619   603  FORMAT(/,F5.2,1PD13.4,D14.4,D13.4,D12.4,2D13.4,D12.4)
  620   6031 FORMAT(/,F5.2,  F13.4,F14.4,F13.4,F12.4,2F13.4,F12.4)
  621   6032 FORMAT(F5.2,27F10.4)
  622   6033 FORMAT(F5.2,27F6.1)
  623   155  FORMAT(/'  pH      Cu(+2)      CuOH(+)  ',
  624       & ' Cu(OH)2(0)  Cu(OH)3(-)  Cu(OH)4(-2)  Cu2(OH)2(+2)  ',
  625       & ' CuSO4(0)    CuEDTA(-2)   CuHEDTA(-)  CuOHEDTA(-3)')
  626   156  FORMAT(/'  pH     CuNTA(-)   Cu(NTA)2(-4) ',
  627       & ' CuOHNTA(-2)   CuCO3(0)   Cu(CO3)2(-2)   CuHCO3(+)   ',
  628       & ' CuCl(+)     CuCl2(0)       CuA2       CuNH3(+2)')
  629   157  FORMAT(/'  pH   Cu(NH3)2(+2)  Cu(NH3)3(+2)   ',
  630       & ' CuAc(+)     CuAc2(0)          SSCu         CuF(+)       T.D.Cu')
  631  C
  632         RETURN
  633         END


WCU  Local Symbols

Name                        Class   Type              Size   Offset

MTLFILE . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . param                             000e
TDCU. . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . local   INTEGER*4            4    0246
J . . . . . . . . . . . . . local   INTEGER*4            4    024a
T . . . . . . . . . . . . . local   REAL*8               8    024e
ISTART. . . . . . . . . . . local   INTEGER*4            4    0256
Y . . . . . . . . . . . . . local   REAL*8             208    025a
LABEL . . . . . . . . . . . local   CHAR*38             38    032a
IEND. . . . . . . . . . . . local   INTEGER*4            4    0350
SKIP. . . . . . . . . . . . local   INTEGER*4            4    0354
PHS . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . W       REAL*8               8    0008
```

```
WCU  Local Symbols

Name                         Class   Type              Size   Offset

F3. . . . . . . . . . . . . . W      REAL*8               8    0010
NUMPH . . . . . . . . . . . . W      INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W      INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W      INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W      CHAR*27             27    0020
```

```
C
      SUBROUTINE WFEII(TDFE2,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE IRON(II) SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDFE2(41)
      REAL*8 X(319,41),Y(19)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF IRON                 '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF IRON            '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF IRON       '
        WRITE(PR,243) LABEL,SLDID
        WRITE(PR,158)
        DO 58 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6071) PHS(I),
     &      LOGFN(X( 72,I)*F),LOGFN(X( 73,I)*F),
     &      LOGFN(X( 74,I)*F),LOGFN(X( 75,I)*F),
     &      LOGFN(X( 76,I)*F),LOGFN(X( 80,I)*F),
     &      LOGFN(X(212,I)*F),LOGFN(X(213,I)*F),
     &      LOGFN(X(214,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/1D0
            WRITE(PR,6071) PHS(I),
     &      X( 72,I)/T,X( 73,I)/T,
     &      X( 74,I)/T,X( 75,I)/T,
     &      X( 76,I)/T,X( 80,I)/T,
     &      X(212,I)/T,X(213,I)/T,
     &      X(214,I)/T
          ELSE
            WRITE(PR,607) PHS(I),
     &      X( 72,I)*F,X( 73,I)*F,
     &      X( 74,I)*F,X( 75,I)*F,
     &      X( 76,I)*F,X( 80,I)*F,
     &      X(212,I)*F,X(213,I)*F,
     &      X(214,I)*F
          END IF
   58   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,243) LABEL,SLDID
        WRITE(PR,159)
        DO 59 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6081) PHS(I),
     &      LOGFN(X(215,I)*F),LOGFN(X(255,I)*F),
     &      LOGFN(X(256,I)*F),LOGFN(X(257,I)*F),
     &      LOGFN(X(258,I)*F),LOGFN(X( 77,I)*F),
     &      LOGFN(X( 78,I)*F),LOGFN(X( 79,I)*F),
     &      LOGFN(X(273,I)*F),LOGFN(TDFE2(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6081) PHS(I),
     &      X(215,I)/T,X(255,I)/T,
     &      X(256,I)/T,X(257,I)/T,
     &      X(258,I)/T,X( 77,I)/T,
     &      X( 78,I)/T,X( 79,I)/T,
     &      X(273,I)/T,TDFE2(I)/T
          ELSE
            WRITE(PR,608) PHS(I),
     &      X(215,I)*F,X(255,I)*F,
     &      X(256,I)*F,X(257,I)*F,
     &      X(258,I)*F,X( 77,I)*F,
     &      X( 78,I)*F,X( 79,I)*F,
     &      X(273,I)*F,TDFE2(I)*F
          END IF
   59   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
  &               RECL=200)
     WRITE(7,242) LABEL
     DO 92 I=1,NUMPH
       IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
         Y(1 )=LOGFN(X( 72,I)*F)
         Y(2 )=LOGFN(X( 73,I)*F)
         Y(3 )=LOGFN(X( 74,I)*F)
         Y(4 )=LOGFN(X( 75,I)*F)
         Y(5 )=LOGFN(X( 76,I)*F)
         Y(6 )=LOGFN(X( 80,I)*F)
         Y(7 )=LOGFN(X(212,I)*F)
         Y(8 )=LOGFN(X(213,I)*F)
         Y(9 )=LOGFN(X(214,I)*F)
         Y(10)=LOGFN(X(215,I)*F)
         Y(11)=LOGFN(X(255,I)*F)
         Y(12)=LOGFN(X(256,I)*F)
         Y(13)=LOGFN(X(257,I)*F)
         Y(14)=LOGFN(X(258,I)*F)
         Y(15)=LOGFN(X( 77,I)*F)
         Y(16)=LOGFN(X( 78,I)*F)
         Y(17)=LOGFN(X( 79,I)*F)
         Y(18)=LOGFN(X(273,I)*F)
         Y(19)=LOGFN(TDFE2(I)*F)
         WRITE(7,6082) PHS(I),(Y(J),J=1,19)
       ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
         WRITE(7,6083) PHS(I),
  &      X( 72,I)/T,X( 73,I)/T,
  &      X( 74,I)/T,X( 75,I)/T,
  &      X( 76,I)/T,X( 80,I)/T,
  &      X(212,I)/T,X(213,I)/T,
  &      X(214,I)/T,
  &      X(215,I)/T,X(255,I)/T,
  &      X(256,I)/T,X(257,I)/T,
  &      X(258,I)/T,X( 77,I)/T,
  &      X( 78,I)/T,X( 79,I)/T,
  &      X(273,I)/T,TDFE2(I)/T
       ELSE
         WRITE(7,6082) PHS(I),
  &      X( 72,I)*F,X( 73,I)*F,
  &      X( 74,I)*F,X( 75,I)*F,
  &      X( 76,I)*F,X( 80,I)*F,
  &      X(212,I)*F,X(213,I)*F,
  &      X(214,I)*F,
  &      X(215,I)*F,X(255,I)*F,
  &      X(256,I)*F,X(257,I)*F,
  &      X(258,I)*F,X( 77,I)*F,
  &      X( 78,I)*F,X( 79,I)*F,
  &      X(273,I)*F,TDFE2(I)*F
       END IF
92   CONTINUE
     CLOSE (UNIT=7)
```

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  776          END IF
  777   C
  778    242   FORMAT(     '     IRON(II) SYSTEM IN ',A38)
  779    243   FORMAT(/T44,'     IRON(II) SYSTEM IN ',A38//T50,'SOLID = ',A27)
  780    607   FORMAT(/,F5.2,1P3D13.4,2(2D13.4,D14.4))
  781    6071  FORMAT(/,F5.2,  3F13.4,2(2F13.4,F14.4))
  782    608   FORMAT(/,F5.2,1PD13.4,D14.4,4D13.4,4D12.4)
  783    6081  FORMAT(/,F5.2,  F13.4,F14.4,4F13.4,4F12.4)
  784    6082  FORMAT(F5.2,19F10.4)
  785    6083  FORMAT(F5.2,19F6.1)
  786    158   FORMAT(/'  pH       Fe(+2)        FeOH(+)    ',
  787         & ' Fe(OH)2(0)   Fe(OH)3(-)   Fe(OH)4(-2)    FeSO4(0)    ',
  788         & ' FeEDTA(-2)   FeHEDTA(-)   FeOHEDTA(-3)')
  789    159   FORMAT(/'  pH  Fe(OH)2EDTA(-4)  FeNTA(-) '
  790         & '  Fe(NTA)2(-4)   FeHNTA(0)   FeOHNTA(-2)  ',
  791         & ' FeHPO4(0)  FeH2PO4(+)    FeAC(+)     SSFe(II) ',
  792         & ' T.D.Fe(II)')
  793   C
  794          RETURN
  795          END


WFEII  Local Symbols

Name                         Class   Type               Size   Offset

MTLFILE . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . param                             000e
TDFE2 . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . local   INTEGER*4             4   0358
J . . . . . . . . . . . . . local   INTEGER*4             4   035c
T . . . . . . . . . . . . . local   REAL*8                8   0360
ISTART. . . . . . . . . . . local   INTEGER*4             4   0368
Y . . . . . . . . . . . . . local   REAL*8              152   036c
LABEL . . . . . . . . . . . local   CHAR*38              38   0404
IEND. . . . . . . . . . . . local   INTEGER*4             4   042a
SKIP. . . . . . . . . . . . local   INTEGER*4             4   042e
PHS . . . . . . . . . . . . OUT     REAL*4              164   98b8
X . . . . . . . . . . . . . OUT     REAL*8           104632   0000
F . . . . . . . . . . . . . W       REAL*8                8   0000
F2. . . . . . . . . . . . . W       REAL*8                8   0008
F3. . . . . . . . . . . . . W       REAL*8                8   0010
NUMPH . . . . . . . . . . . W       INTEGER*4             4   0018
NPL . . . . . . . . . . . . W       INTEGER*4             4   001c
PR. . . . . . . . . . . . . W       INTEGER*4             4   003c
SLDID . . . . . . . . . . . W       CHAR*27              27   0020
```

```
Global Symbols

Name                          Class   Type             Size   Offset

LOGFN . . . . . . . . . . . . extern  REAL*8            ***      ***
OUT . . . . . . . . . . . . . common  ***            104796     0000
W . . . . . . . . . . . . . . common  ***                64     0000
WCD . . . . . . . . . . . . . FSUBRT  ***               ***     3e13
WCU . . . . . . . . . . . . . FSUBRT  ***               ***     7c91
WFEII . . . . . . . . . . . . FSUBRT  ***               ***     b8e4
WFOOT . . . . . . . . . . . . extern  ***               ***      ***
WHEAD . . . . . . . . . . . . extern  ***               ***      ***
WHGII . . . . . . . . . . . . FSUBRT  ***               ***     0000

Code size = e3b8 (58296)
Data size = 0915 (2325)
Bss size  = 0432 (1074)

No errors detected
```

```
      SUBROUTINE WMNII(TDMN2,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE MANGANESE(II) SUBSYSTEM AQUEOUS SPECIES REP
C
      REAL*4 PHS(41),TDMN2(41)
      REAL*8 X(319,41),Y(18)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF MANGANESE            '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF MANGANESE       '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF MANGANESE  '
        WRITE(PR,245) LABEL,SLDID
        WRITE(PR,160)
        DO 60 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6091) PHS(I),
     &      LOGFN(X(179,I)*F)   ,LOGFN(X(115,I)*F),
     &      LOGFN(X(116,I)*F)   ,LOGFN(X(111,I)*F*F2),
     &      LOGFN(X(112,I)*F*F2),LOGFN(X(114,I)*F),
     &      LOGFN(X(216,I)*F)   ,LOGFN(X(217,I)*F),
     &      LOGFN(X(259,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6091) PHS(I),
     &      X(179,I)/T   ,X(115,I)/T,
     &      X(116,I)/T   ,X(111,I)/T*F2,
     &      X(112,I)/T*F2,X(114,I)/T,
     &      X(216,I)/T   ,X(217,I)/T,
     &      X(259,I)/T
          ELSE
            WRITE(PR,609) PHS(I),
     &      X(179,I)*F   ,X(115,I)*F,
     &      X(116,I)*F   ,X(111,I)*F*F2,
     &      X(112,I)*F*F2,X(114,I)*F,
     &      X(216,I)*F   ,X(217,I)*F,
     &      X(259,I)*F
          END IF
   60   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,245) LABEL,SLDID
        WRITE(PR,161)
        DO 61 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6101) PHS(I),
     &      LOGFN(X(260,I)*F),LOGFN(X(113,I)*F),
     &      LOGFN(X(117,I)*F),LOGFN(X(119,I)*F),
     &      LOGFN(X(120,I)*F),LOGFN(X(121,I)*F),
     &      LOGFN(X(118,I)*F),LOGFN(X(275,I)*F),
     &      LOGFN(TDMN2(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6101) PHS(I),
     &      X(260,I)/T,X(113,I)/T,
     &      X(117,I)/T,X(119,I)/T,
     &      X(120,I)/T,X(121,I)/T,
     &      X(118,I)/T,X(275,I)/T,
     &      TDMN2(I)/T
          ELSE
            WRITE(PR,610) PHS(I),
     &      X(260,I)*F,X(113,I)*F,
     &      X(117,I)*F,X(119,I)*F,
     &      X(120,I)*F,X(121,I)*F,
     &      X(118,I)*F,X(275,I)*F,
     &      TDMN2(I)*F
          END IF
   61   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=190)
        WRITE(7,244) LABEL
        DO 93 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(179,I)*F)
            Y(2 )=LOGFN(X(115,I)*F)
            Y(3 )=LOGFN(X(116,I)*F)
            Y(4 )=LOGFN(X(111,I)*F*F2)
            Y(5 )=LOGFN(X(112,I)*F*F2)
            Y(6 )=LOGFN(X(114,I)*F)
            Y(7 )=LOGFN(X(216,I)*F)
            Y(8 )=LOGFN(X(217,I)*F)
            Y(9 )=LOGFN(X(259,I)*F)
            Y(10)=LOGFN(X(260,I)*F)
            Y(11)=LOGFN(X(113,I)*F)
            Y(12)=LOGFN(X(117,I)*F)
            Y(13)=LOGFN(X(119,I)*F)
            Y(14)=LOGFN(X(120,I)*F)
            Y(15)=LOGFN(X(121,I)*F)
            Y(16)=LOGFN(X(118,I)*F)
            Y(17)=LOGFN(X(275,I)*F)
            Y(18)=LOGFN(TDMN2(I)*F)
            WRITE(7,6102) PHS(I),(Y(J),J=1,18)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6103) PHS(I),
     &      X(179,I)/T   ,X(115,I)/T,
     &      X(116,I)/T   ,X(111,I)/T*F2,
     &      X(112,I)/T*F2,X(114,I)/T,
     &      X(216,I)/T   ,X(217,I)/T,
     &      X(259,I)/T,
     &      X(260,I)/T,X(113,I)/T,
     &      X(117,I)/T,X(119,I)/T,
     &      X(120,I)/T,X(121,I)/T,
     &      X(118,I)/T,X(275,I)/T,
     &      TDMN2(I)/T
          ELSE
            WRITE(7,6102) PHS(I),
     &      X(179,I)*F   ,X(115,I)*F,
     &      X(116,I)*F   ,X(111,I)*F*F2,
     &      X(112,I)*F*F2,X(114,I)*F,
     &      X(216,I)*F   ,X(217,I)*F,
     &      X(259,I)*F,
     &      X(260,I)*F,X(113,I)*F,
     &      X(117,I)*F,X(119,I)*F,
     &      X(120,I)*F,X(121,I)*F,
     &      X(118,I)*F,X(275,I)*F,
     &      TDMN2(I)*F
          END IF
   93   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 244  FORMAT(     ' MANGANESE SYSTEM IN ',A38)
 245  FORMAT(/T44,' MANGANESE SYSTEM IN ',A38//T50,'SOLID = ',A27)
 609  FORMAT(/,F5.2,1P3D13.4,3D14.4,3D13.4)
 6091 FORMAT(/,F5.2,  3F13.4,3F14.4,3F13.4)
 610  FORMAT(/,F5.2,1P2D14.4,2D13.4,D14.4,D15.4,3D14.4)
 6101 FORMAT(/,F5.2,  2F14.4,2F13.4,F14.4,F15.4,3F14.4)
 6102 FORMAT(F5.2,18F10.4)
 6103 FORMAT(F5.2,18F6.1)
 160  FORMAT(/'  pH       Mn(+2)       MnOH(+)     ',
     & ' Mn(OH)4(-2)    Mn2OH(+3)    Mn2(OH)3(+)    ',
     & ' MnSO4(0)    MnEDTA(-2)   MnHEDTA(-)    MnNTA(-)')
```

```
Line#  Source Line        Microsoft FORTRAN Optimizing Compiler Version 5.00

  154   161  FORMAT(/'  pH     Mn(NTA)2(-4)     MnHCO3(+)     ',
  155       & ' MnCl(+)      MnNH3(+2)    Mn(NH3)2(+2)   ',
  156       & ' Mn(NH3)3(+2)        MnAC(+)          SSMn       ',
  157       & '   T.D.Mn(II)')
  158   C
  159          RETURN
  160          END


WMNII  Local Symbols

Name                         Class    Type                Size    Offset

MTLFILE . . . . . . . . . . param                                 0006
UNIT. . . . . . . . . . . . param                                 000a
T1. . . . . . . . . . . . . param                                 000e
TDMN2 . . . . . . . . . . . param                                 0012
I . . . . . . . . . . . . . local    INTEGER*4              4     0002
J . . . . . . . . . . . . . local    INTEGER*4              4     0006
T . . . . . . . . . . . . . local    REAL*8                 8     000a
ISTART. . . . . . . . . . . local    INTEGER*4              4     0012
Y . . . . . . . . . . . . . local    REAL*8               144     0016
LABEL . . . . . . . . . . . local    CHAR*38               38     00a6
IEND. . . . . . . . . . . . local    INTEGER*4              4     00cc
SKIP. . . . . . . . . . . . local    INTEGER*4              4     00d0
PHS . . . . . . . . . . . . OUT      REAL*4               164     98b8
X . . . . . . . . . . . . . OUT      REAL*8            104632     0000
F . . . . . . . . . . . . . W        REAL*8                 8     0000
F2. . . . . . . . . . . . . W        REAL*8                 8     0008
F3. . . . . . . . . . . . . W        REAL*8                 8     0010
NUMPH . . . . . . . . . . . W        INTEGER*4              4     0018
NPL . . . . . . . . . . . . W        INTEGER*4              4     001c
PR. . . . . . . . . . . . . W        INTEGER*4              4     003c
SLDID . . . . . . . . . . . W        CHAR*27               27     0020

    161   C
    162          SUBROUTINE WZN(TDZN,T1,UNIT,MTLFILE)
    163   C      THIS BLOCK WRITES THE ZINC SUBSYSTEM AQUEOUS SPECIES REPORT.
    164   C
    165          REAL*4 PHS(41),TDZN(41)
    166          REAL*8 X(319,41),Y(30)
    167          COMMON /OUT/X,PHS
    168          INTEGER I,J,ISTART,IEND,SKIP,UNIT
    169          REAL*8 F,F2,F3,T,T1
    170          INTEGER NUMPH,NPL,PR
    171          CHARACTER LABEL*38,SLDID*27
    172          CHARACTER MTLFILE*15
    173          REAL*8 LOGFN
    174          COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
    175   C
    176          ISTART=1
```

```fortran
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MINO(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF ZINC                 '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF ZINC            '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF ZINC       '
        WRITE(PR,247) LABEL,SLDID
        WRITE(PR,162)
        DO 62 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6391) PHS(I),
     &      LOGFN(X(180,I)*F)   ,LOGFN(X(122,I)*F),
     &      LOGFN(X(123,I)*F)   ,LOGFN(X(124,I)*F),
     &      LOGFN(X(125,I)*F*F2),LOGFN(X(128,I)*F),
     &      LOGFN(X(129,I)*F)   ,LOGFN(X(130,I)*F),
     &      LOGFN(X(218,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6391) PHS(I),
     &      X(180,I)/T   ,X(122,I)/T,
     &      X(123,I)/T   ,X(124,I)/T,
     &      X(125,I)/T*F2,X(128,I)/T,
     &      X(129,I)/T   ,X(130,I)/T,
     &      X(218,I)/T
          ELSE
            WRITE(PR,639) PHS(I),
     &      X(180,I)*F   ,X(122,I)*F,
     &      X(123,I)*F   ,X(124,I)*F,
     &      X(125,I)*F*F2,X(128,I)*F,
     &      X(129,I)*F   ,X(130,I)*F,
     &      X(218,I)*F
          END IF
   62   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,247) LABEL,SLDID
        WRITE(PR,163)
        DO 63 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6121) PHS(I),
     &      LOGFN(X(219,I)*F),LOGFN(X(220,I)*F),
     &      LOGFN(X(261,I)*F),LOGFN(X(262,I)*F),
     &      LOGFN(X(263,I)*F),LOGFN(X(131,I)*F),
     &    LOGFN(X(132,I)*F),LOGFN(X(133,I)*F),
     &    LOGFN(X(137,I)*F),LOGFN(X(138,I)*F)
        ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
          WRITE(PR,6121) PHS(I),
     &    X(219,I)/T,X(220,I)/T,
     &    X(261,I)/T,X(262,I)/T,
     &    X(263,I)/T,X(131,I)/T,
     &    X(132,I)/T,X(133,I)/T,
     &    X(137,I)/T,X(138,I)/T
        ELSE
          WRITE(PR,612) PHS(I),
     &    X(219,I)*F,X(220,I)*F,
     &    X(261,I)*F,X(262,I)*F,
     &    X(263,I)*F,X(131,I)*F,
     &    X(132,I)*F,X(133,I)*F,
     &    X(137,I)*F,X(138,I)*F
        END IF
   63 CONTINUE
      CALL WFOOT(SKIP)
C
      CALL WHEAD
      WRITE(PR,247) LABEL,SLDID
      WRITE(PR,164)
      DO 64 I=ISTART,IEND
        IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
          WRITE(PR,6131) PHS(I),
     &    LOGFN(X(139,I)*F),LOGFN(X(126,I)*F),
     &    LOGFN(X(127,I)*F),LOGFN(X(134,I)*F),
     &    LOGFN(X(135,I)*F),LOGFN(X(136,I)*F),
     &    LOGFN(X(188,I)*F),LOGFN(X(140,I)*F),
     &    LOGFN(X(276,I)*F),LOGFN(X(316,I)*F)
        ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
          WRITE(PR,6131) PHS(I),
     &    X(139,I)/T,X(126,I)/T,
     &    X(127,I)/T,X(134,I)/T,
     &    X(135,I)/T,X(136,I)/T,
     &    X(188,I)/T,X(140,I)/T,
     &    X(276,I)/T,X(316,I)/T
        ELSE
          WRITE(PR,613) PHS(I),
     &    X(139,I)*F,X(126,I)*F,
     &    X(127,I)*F,X(134,I)*F,
     &    X(135,I)*F,X(136,I)*F,
     &    X(188,I)*F,X(140,I)*F,
     &    X(276,I)*F,X(316,I)*F
        END IF
   64 CONTINUE
      CALL WFOOT(SKIP)
C
      CALL WHEAD
      WRITE(PR,247) LABEL,SLDID
```

```
        WRITE(PR,1641)
        DO I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6121) PHS(I),
     &      LOGFN(TDZN(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6121) PHS(I),
     &      TDZN(I)/T
          ELSE
            WRITE(PR,612) PHS(I),
     &      TDZN(I)*F
          END IF
        END DO
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=310)
        WRITE(7,246) LABEL
        DO 94 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(180,I)*F)
            Y(2 )=LOGFN(X(122,I)*F)
            Y(3 )=LOGFN(X(123,I)*F)
            Y(4 )=LOGFN(X(124,I)*F)
            Y(5 )=LOGFN(X(125,I)*F*F2)
            Y(6 )=LOGFN(X(128,I)*F)
            Y(7 )=LOGFN(X(129,I)*F)
            Y(8 )=LOGFN(X(130,I)*F)
            Y(9 )=LOGFN(X(218,I)*F)
            Y(10)=LOGFN(X(219,I)*F)
            Y(11)=LOGFN(X(220,I)*F)
            Y(12)=LOGFN(X(261,I)*F)
            Y(13)=LOGFN(X(262,I)*F)
            Y(14)=LOGFN(X(263,I)*F)
            Y(15)=LOGFN(X(131,I)*F)
            Y(16)=LOGFN(X(132,I)*F)
            Y(17)=LOGFN(X(133,I)*F)
            Y(18)=LOGFN(X(137,I)*F)
            Y(19)=LOGFN(X(138,I)*F)
            Y(20)=LOGFN(X(139,I)*F)
            Y(21)=LOGFN(X(126,I)*F)
            Y(22)=LOGFN(X(127,I)*F)
            Y(23)=LOGFN(X(134,I)*F)
            Y(24)=LOGFN(X(135,I)*F)
            Y(25)=LOGFN(X(136,I)*F)
            Y(26)=LOGFN(X(188,I)*F)
            Y(27)=LOGFN(X(140,I)*F)
            Y(28)=LOGFN(X(276,I)*F)
            Y(29)=LOGFN(X(316,I)*F)
            Y(30)=LOGFN(TDZN(I)*F)
            WRITE(7,6132) PHS(I),(Y(J),J=1,30)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6133) PHS(I),
     &      X(180,I)/T   ,X(122,I)/T,
     &      X(123,I)/T   ,X(124,I)/T,
     &      X(125,I)/T*F2,X(128,I)/T,
     &      X(129,I)/T   ,X(130,I)/T,
     &      X(218,I)/T,
     &      X(219,I)/T,X(220,I)/T,
     &      X(261,I)/T,X(262,I)/T,
     &      X(263,I)/T,X(131,I)/T,
     &      X(132,I)/T,X(133,I)/T,
     &      X(137,I)/T,X(138,I)/T,
     &      X(139,I)/T,X(126,I)/T,
     &      X(127,I)/T,X(134,I)/T,
     &      X(135,I)/T,X(136,I)/T,
     &      X(188,I)/T,X(140,I)/T,
     &      X(276,I)/T,X(316,I)/T,
     &      TDZN(I)/T
          ELSE
            WRITE(7,6132) PHS(I),
     &      X(180,I)*F   ,X(122,I)*F,
     &      X(123,I)*F   ,X(124,I)*F,
     &      X(125,I)*F*F2,X(128,I)*F,
     &      X(129,I)*F   ,X(130,I)*F,
     &      X(218,I)*F,
     &      X(219,I)*F,X(220,I)*F,
     &      X(261,I)*F,X(262,I)*F,
     &      X(263,I)*F,X(131,I)*F,
     &      X(132,I)*F,X(133,I)*F,
     &      X(137,I)*F,X(138,I)*F,
     &      X(139,I)*F,X(126,I)*F,
     &      X(127,I)*F,X(134,I)*F,
     &      X(135,I)*F,X(136,I)*F,
     &      X(188,I)*F,X(140,I)*F,
     &      X(276,I)*F,X(316,I)*F,
     &      TDZN(I)*F
          END IF
   94   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 246  FORMAT(    '       ZINC SYSTEM IN ',A38)
 247  FORMAT(/T44,'      ZINC SYSTEM IN ',A38//T50,' SOLID = ',A27)
 639  FORMAT(/,F5.2,1P6D13.4,D14.4,D15.4,D14.4)
 6391 FORMAT(/,F5.2,  6F13.4,F14.4,F15.4,F14.4)
 612  FORMAT(/,F5.2,1PD12.4,5D13.4,3D12.4,D13.4)
 6121 FORMAT(/,F5.2,  F12.4,5F13.4,3F12.4,F13.4)
 613  FORMAT(/,F5.2,1PD14.4,D13.4,3D12.4,2D13.4,3D12.4)
```

```
  381   6131 FORMAT(/,F5.2,  F14.4,F13.4,3F12.4,2F13.4,3F12.4)
  382   6132 FORMAT(F5.2,30F10.4)
  383   6133 FORMAT(F5.2,30F6.1)
  384   162  FORMAT(/'  pH       Zn(+2)        ZnOH(+)    ',
  385       & ' Zn(OH)2(0)   Zn(OH)3(-)    Zn2OH(+3)     ZnSO4(0)    ',
  386       & ' Zn(SO4)2(-2)   Zn(SO4)3(-4)   ZnEDTA(-2)')
  387   163  FORMAT(/'  pH    ZnHEDTA(-)  ZnOHEDTA(-3)',
  388       & '    ZnNTA(-)    Zn(NTA)2(-4)  ZnOHNTA(-2)     ZnCl(+) ',
  389       & '    ZnCl2(0)     ZnCl3(-)     ZnNH3(+2) Zn(NH3)2(+2)')
  390   164  FORMAT(/'  pH     Zn(NH3)3(+2)   ZnHPO4(0)',
  391       & '   ZnH2PO4(+)     ZnAC(+)   Zn(AC)2(0)    Zn(AC)3(-)',
  392       & '    Zn(HS)2(0)    ZnHSOH(0)      SSZn        ZnF(+)')
  393   1641 FORMAT(/'  pH    T.D.Zn')
  394   C
  395          RETURN
  396          END


WZN  Local Symbols

Name                          Class   Type              Size   Offset

MTLFILE . . . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . . . param                             000e
TDZN. . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local   INTEGER*4            4    00d4
J . . . . . . . . . . . . . . local   INTEGER*4            4    00d8
T . . . . . . . . . . . . . . local   REAL*8               8    00dc
ISTART. . . . . . . . . . . . local   INTEGER*4            4    00e4
Y . . . . . . . . . . . . . . local   REAL*8             240    00e8
LABEL . . . . . . . . . . . . local   CHAR*38             38    01d8
IEND. . . . . . . . . . . . . local   INTEGER*4            4    01fe
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    0202
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020

  397   C
  398          SUBROUTINE WK(TDK,T1,UNIT,MTLFILE)
  399   C      THIS BLOCK WRITES THE POTASSIUM SUBSYSTEM AQUEOUS SPECIES REPORT.
  400   C
  401          REAL*4 PHS(41),TDK(41)
  402          REAL*8 X(319,41),Y(8)
  403          COMMON /OUT/X,PHS
```

```
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MINO(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES            '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES       '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF POTASSIUM             '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF POTASSIUM        '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF POTASSIUM   '
        WRITE(PR,249) LABEL,SLDID
        WRITE(PR,165)
        DO 65 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6401) PHS(I),
     &      LOGFN(X(182,I)*F),LOGFN(X(145,I)*F),
     &      LOGFN(X(147,I)*F),LOGFN(X(221,I)*F),
     &      LOGFN(X(148,I)*F),LOGFN(X(146,I)*F),
     &      LOGFN(X(277,I)*F),LOGFN(TDK(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6401) PHS(I),
     &      X(182,I)/T,X(145,I)/T,
     &      X(147,I)/T,X(221,I)/T,
     &      X(148,I)/T,X(146,I)/T,
     &      X(277,I)/T,TDK(I)/T
          ELSE
            WRITE(PR,640) PHS(I),
     &      X(182,I)*F,X(145,I)*F,
     &      X(147,I)*F,X(221,I)*F,
     &      X(148,I)*F,X(146,I)*F,
     &      X(277,I)*F,TDK(I)*F
          END IF
   65   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
```

```
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=90)
        WRITE(7,248) LABEL
        DO 95 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(182,I)*F)
            Y(2 )=LOGFN(X(145,I)*F)
            Y(3 )=LOGFN(X(147,I)*F)
            Y(4 )=LOGFN(X(221,I)*F)
            Y(5 )=LOGFN(X(148,I)*F)
            Y(6 )=LOGFN(X(146,I)*F)
            Y(7 )=LOGFN(X(277,I)*F)
            Y(8 )=LOGFN(TDK(I)*F)
            WRITE(7,6402) PHS(I),(Y(J),J=1,8)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6403) PHS(I),
     &      X(182,I)/T,X(145,I)/T,
     &      X(147,I)/T,X(221,I)/T,
     &      X(148,I)/T,X(146,I)/T,
     &      X(277,I)/T,TDK(I)/T
          ELSE
            WRITE(7,6402) PHS(I),
     &      X(182,I)*F,X(145,I)*F,
     &      X(147,I)*F,X(221,I)*F,
     &      X(148,I)*F,X(146,I)*F,
     &      X(277,I)*F,TDK(I)*F
          END IF
   95   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 248  FORMAT(     '   POTASSIUM SYSTEM IN ',A38)
 249  FORMAT(/T44,'   POTASSIUM SYSTEM IN ',A38//T50,'SOLID = ',A27)
 640  FORMAT(/,F5.2,1P10D13.4)
 6401 FORMAT(/,F5.2,  10F13.4)
 6402 FORMAT(F5.2,8F10.4)
 6403 FORMAT(F5.2,8F6.1)
 165  FORMAT(/'  pH       K(+)        KOH(0)       ',
     & ' KS04(-)     KEDTA(-3)     KC1(0)      KHP04(-)   ',
     & '      SSHK       T.D.K')
C
      RETURN
      END
```

```
WK  Local Symbols

Name                       Class   Type              Size   Offset

MTLFILE . . . . . . . . . param                              0006
UNIT. . . . . . . . . . . param                              000a
```

```
WK  Local Symbols

Name                          Class   Type              Size   Offset

T1. . . . . . . . . . . . . . param                             000e
TDK . . . . . . . . . . . . . param                             0012
I . . . . . . . . . . . . . . local   INTEGER*4            4    0206
J . . . . . . . . . . . . . . local   INTEGER*4            4    020a
T . . . . . . . . . . . . . . local   REAL*8               8    020e
ISTART. . . . . . . . . . . . local   INTEGER*4            4    0216
Y . . . . . . . . . . . . . . local   REAL*8              64    021a
LABEL . . . . . . . . . . . . local   CHAR*38             38    025a
IEND. . . . . . . . . . . . . local   INTEGER*4            4    0280
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    0284
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020
```

```
C
      SUBROUTINE WCO(TDCO,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE COBALT SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDCO(41)
      REAL*8 X(319,41),Y(19)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES                 '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES            '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF COBALT                     '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF COBALT                '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF COBALT      '
        WRITE(PR,251) LABEL,SLDID
        WRITE(PR,166)
        DO 66 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6141) PHS(I),
     &      LOGFN(X(222,I)*F),LOGFN(X(225,I)*F),
     &      LOGFN(X(226,I)*F),LOGFN(X(227,I)*F),
     &      LOGFN(X(228,I)*F),LOGFN(X(229,I)*F*F2),
     &      LOGFN(X(230,I)*F),LOGFN(X(224,I)*F),
     &      LOGFN(X(223,I)*F),LOGFN(X(264,I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1D0
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,6141) PHS(I),
     &      X(222,I)/T,X(225,I)/T,
     &      X(226,I)/T,X(227,I)/T,
     &      X(228,I)/T,X(229,I)/T*F2,
     &      X(230,I)/T,X(224,I)/T,
     &      X(223,I)/T,X(264,I)/T
          ELSE
            WRITE(PR,614) PHS(I),
     &      X(222,I)*F,X(225,I)*F,
     &      X(226,I)*F,X(227,I)*F,
     &      X(228,I)*F,X(229,I)*F*F2,
     &      X(230,I)*F,X(224,I)*F,
     &      X(223,I)*F,X(264,I)*F
          END IF
   66   CONTINUE
        SKIP=2*(NPL+ISTART-IEND)
        CALL WFOOT(SKIP)
C
        CALL WHEAD
        WRITE(PR,251) LABEL,SLDID
        WRITE(PR,167)
        DO 67 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6151) PHS(I),
     &      LOGFN(X(266,I)*F),LOGFN(X(265,I)*F),
     &      LOGFN(X(231,I)*F),LOGFN(X(232,I)*F),
     &      LOGFN(X(233,I)*F),LOGFN(X(234,I)*F),
     &      LOGFN(X(235,I)*F),LOGFN(X(278,I)*F),
     &      LOGFN(TDCO(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6151) PHS(I),
     &      X(266,I)/T,X(265,I)/T,
     &      X(231,I)/T,X(232,I)/T,
     &      X(233,I)/T,X(234,I)/T,
     &      X(235,I)/T,X(278,I)/T,
     &      TDCO(I)/T
          ELSE
```

```
            WRITE(PR,615) PHS(I),
     &      X(266,I)*F,X(265,I)*F,
     &      X(231,I)*F,X(232,I)*F,
     &      X(233,I)*F,X(234,I)*F,
     &      X(235,I)*F,X(278,I)*F,
     &      TDCO(I)*F
          END IF
   67   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=200)
        WRITE(7,250) LABEL
        DO 96 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(222,I)*F)
            Y(2 )=LOGFN(X(225,I)*F)
            Y(3 )=LOGFN(X(226,I)*F)
            Y(4 )=LOGFN(X(227,I)*F)
            Y(5 )=LOGFN(X(228,I)*F)
            Y(6 )=LOGFN(X(229,I)*F*F2)
            Y(7 )=LOGFN(X(230,I)*F)
            Y(8 )=LOGFN(X(224,I)*F)
            Y(9 )=LOGFN(X(223,I)*F)
            Y(10)=LOGFN(X(264,I)*F)
            Y(11)=LOGFN(X(266,I)*F)
            Y(12)=LOGFN(X(265,I)*F)
            Y(13)=LOGFN(X(231,I)*F)
            Y(14)=LOGFN(X(232,I)*F)
            Y(15)=LOGFN(X(233,I)*F)
            Y(16)=LOGFN(X(234,I)*F)
            Y(17)=LOGFN(X(235,I)*F)
            Y(18)=LOGFN(X(278,I)*F)
            Y(19)=LOGFN(TDCO(I)*F)
            WRITE(7,6152) PHS(I),(Y(J),J=1,19)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6153) PHS(I),
     &      X(222,I)/T,X(225,I)/T,
     &      X(226,I)/T,X(227,I)/T,
     &      X(228,I)/T,X(229,I)/T*F2,
     &      X(230,I)/T,X(224,I)/T,
     &      X(223,I)/T,
     &      X(264,I)/T,X(266,I)/T,
     &      X(265,I)/T,X(231,I)/T,
     &      X(232,I)/T,X(233,I)/T,
     &      X(234,I)/T,X(235,I)/T,
     &      X(278,I)/T,TDCO(I)/T
          ELSE
```

```
            WRITE(7,6152) PHS(I),
     &      X(222,I)*F,X(225,I)*F,
     &      X(226,I)*F,X(227,I)*F,
     &      X(228,I)*F,X(229,I)*F*F2,
     &      X(230,I)*F,X(224,I)*F,
     &      X(223,I)*F,
     &      X(264,I)*F,X(266,I)*F,
     &      X(265,I)*F,X(231,I)*F,
     &      X(232,I)*F,X(233,I)*F,
     &      X(234,I)*F,X(235,I)*F,
     &      X(278,I)*F,TDCO(I)*F
          END IF
   96   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 250  FORMAT(     '     COBALT SYSTEM IN ',A38)
 251  FORMAT(/T44,'     COBALT SYSTEM IN ',A38//T50,'SOLID = ',A27)
 614  FORMAT(/,F5.2,1P2D12.4,7D13.4,D12.4)
 6141 FORMAT(/,F5.2,  2F12.4,7F13.4,F12.4)
 615  FORMAT(/,F5.2,1PD13.4,2(D15.4,2D14.4),D13.4,D12.4)
 6151 FORMAT(/,F5.2,  F13.4,2(F15.4,2F14.4),F13.4,F12.4)
 6152 FORMAT(F5.2,19F10.4)
 6153 FORMAT(F5.2,19F6.1)
 166  FORMAT(/'  pH      Co(+2)      CoOH(+)',
     & '   Co(OH)2(0)   Co(OH)3(-)  Co(OH)4(-2)    Co2OH(+3)',
     & '    CoSO4(0)    CoEDTA(-2)  CoHEDTA(-)   CoNTA(-)')
 167  FORMAT(/'  pH    CoOHNTA(-2)   ',
     & ' Co(NTA)2(-4)    CoNH3(+2)   Co(NH3)2(+2)  ',
     & ' Co(NH3)3(+2)    CoHPO4(0)      CoAC(+)    ',
     & '     SSCo        T.D.Co')
C
      RETURN
      END
```

```
WCO  Local Symbols

Name                      Class   Type              Size   Offset

MTLFILE . . . . . . . . . param                             0006
UNIT. . . . . . . . . . . param                             000a
T1. . . . . . . . . . . . param                             000e
TDCO. . . . . . . . . . . param                             0012
I . . . . . . . . . . . . local   INTEGER*4            4    0288
J . . . . . . . . . . . . local   INTEGER*4            4    028c
T . . . . . . . . . . . . local   REAL*8               8    0290
ISTART. . . . . . . . . . local   INTEGER*4            4    0298
Y . . . . . . . . . . . . local   REAL*8             152    029c
LABEL . . . . . . . . . . local   CHAR*38             38    0334
IEND. . . . . . . . . . . local   INTEGER*4            4    035a
```

```
WCO  Local Symbols

Name                          Class   Type              Size   Offset

SKIP. . . . . . . . . . . . . local   INTEGER*4            4    035e
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
```

```
C
      SUBROUTINE WAL(TDAL,T1,UNIT,MTLFILE)
C     THIS BLOCK WRITES THE ALUMINUM SUBSYSTEM AQUEOUS SPECIES REPORT.
C
      REAL*4 PHS(41),TDAL(41)
      REAL*8 X(319,41),Y(26)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER MTLFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MINO(IEND+NPL,NUMPH)
C
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF ALUMINUM             '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF ALUMINUM        '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF ALUMINUM   '
        WRITE(PR,253) LABEL,SLDID
        WRITE(PR,171)
        DO 97 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6421) PHS(I),
     &      LOGFN(X(284,I)*F),   LOGFN(X(285,I)*F),
     &      LOGFN(X(286,I)*F),   LOGFN(X(287,I)*F),
     &      LOGFN(X(288,I)*F),   LOGFN(X(289,I)*F*F2),
  695       &      LOGFN(X(290,I)*F*F3),LOGFN(X(291,I)*F),
  696       &      LOGFN(X(292,I)*F)
  697             ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
  698               T=1D0
  699               IF (T1.NE.0D0) T=T1/100
  700               WRITE(PR,6421) PHS(I),
  701       &      X(284,I)/T,   X(285,I)/T,
  702       &      X(286,I)/T,   X(287,I)/T,
  703       &      X(288,I)/T,   X(289,I)/T*F2,
  704       &      X(290,I)/T*F3,X(291,I)/T,
  705       &      X(292,I)/T
  706             ELSE
  707               WRITE(PR,642) PHS(I),
  708       &      X(284,I)*F,   X(285,I)*F,
  709       &      X(286,I)*F,   X(287,I)*F,
  710       &      X(288,I)*F,   X(289,I)*F*F2,
  711       &      X(290,I)*F*F3,X(291,I)*F,
  712       &      X(292,I)*F
  713             END IF
  714      97   CONTINUE
  715           SKIP=2*(NPL+ISTART-IEND)
  716           CALL WFOOT(SKIP)
  717  C
  718           CALL WHEAD
  719           WRITE(PR,253) LABEL,SLDID
  720           WRITE(PR,172)
  721           DO 98 I=ISTART,IEND
  722             IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
  723               WRITE(PR,6431) PHS(I),
  724       &      LOGFN(X(293,I)*F),LOGFN(X(294,I)*F),
  725       &      LOGFN(X(295,I)*F),LOGFN(X(296,I)*F),
  726       &      LOGFN(X(297,I)*F),LOGFN(X(298,I)*F),
  727       &      LOGFN(X(299,I)*F),LOGFN(X(300,I)*F),
  728       &      LOGFN(X(301,I)*F)
  729             ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
  730               WRITE(PR,6431) PHS(I),
  731       &      X(293,I)/T,X(294,I)/T,
  732       &      X(295,I)/T,X(296,I)/T,
  733       &      X(297,I)/T,X(298,I)/T,
  734       &      X(299,I)/T,X(300,I)/T,
  735       &      X(301,I)/T
  736             ELSE
  737               WRITE(PR,643) PHS(I),
  738       &      X(293,I)*F,X(294,I)*F,
  739       &      X(295,I)*F,X(296,I)*F,
  740       &      X(297,I)*F,X(298,I)*F,
  741       &      X(299,I)*F,X(300,I)*F,
  742       &      X(301,I)*F
  743             END IF
  744      98   CONTINUE
  745           CALL WFOOT(SKIP)
```

```
C
        CALL WHEAD
        WRITE(PR,253) LABEL,SLDID
        WRITE(PR,173)
        DO 100 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,6441) PHS(I),
     &      LOGFN(X(302,I)*F),LOGFN(X(303,I)*F),
     &      LOGFN(X(304,I)*F),LOGFN(X(305,I)*F),
     &      LOGFN(X(308,I)*F),LOGFN(X(309,I)*F),
     &      LOGFN(X(310,I)*F),LOGFN(TDAL(I)*F)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(PR,6441) PHS(I),
     &      X(302,I)/T,X(303,I)/T,
     &      X(304,I)/T,X(305,I)/T,
     &      X(308,I)/T,X(309,I)/T,
     &      X(310,I)/T,TDAL(I)/T
          ELSE
            WRITE(PR,644) PHS(I),
     &      X(302,I)*F,X(303,I)*F,
     &      X(304,I)*F,X(305,I)*F,
     &      X(308,I)*F,X(309,I)*F,
     &      X(310,I)*F,TDAL(I)*F
          END IF
  100   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
      IF (MTLFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=MTLFILE,STATUS='UNKNOWN',
     &               RECL=300)
        WRITE(7,252) LABEL
        DO 101 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(284,I)*F)
            Y(2 )=LOGFN(X(285,I)*F)
            Y(3 )=LOGFN(X(286,I)*F)
            Y(4 )=LOGFN(X(287,I)*F)
            Y(5 )=LOGFN(X(288,I)*F)
            Y(6 )=LOGFN(X(289,I)*F*F2)
            Y(7 )=LOGFN(X(290,I)*F*F3)
            Y(8 )=LOGFN(X(291,I)*F)
            Y(9 )=LOGFN(X(292,I)*F)
            Y(10)=LOGFN(X(293,I)*F)
            Y(11)=LOGFN(X(294,I)*F)
            Y(12)=LOGFN(X(295,I)*F)
            Y(13)=LOGFN(X(296,I)*F)
            Y(14)=LOGFN(X(297,I)*F)
            Y(15)=LOGFN(X(298,I)*F)
            Y(16)=LOGFN(X(299,I)*F)
            Y(17)=LOGFN(X(300,I)*F)
            Y(18)=LOGFN(X(301,I)*F)
            Y(19)=LOGFN(X(302,I)*F)
            Y(20)=LOGFN(X(303,I)*F)
            Y(21)=LOGFN(X(304,I)*F)
            Y(22)=LOGFN(X(305,I)*F)
            Y(23)=LOGFN(X(308,I)*F)
            Y(24)=LOGFN(X(309,I)*F)
            Y(25)=LOGFN(X(310,I)*F)
            Y(26)=LOGFN(TDAL(I)*F)
            WRITE(7,6432) PHS(I),(Y(J),J=1,26)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,6433) PHS(I),
     &      X(284,I)/T,X(285,I)/T,
     &      X(286,I)/T,X(287,I)/T,
     &      X(288,I)/T,X(289,I)/T*F2,
     &      X(290,I)/T*F3,X(291,I)/T,
     &      X(292,I)/T,X(293,I)/T,
     &      X(294,I)/T,X(295,I)/T,
     &      X(296,I)/T,X(297,I)/T,
     &      X(298,I)/T,X(299,I)/T,
     &      X(300,I)/T,X(301,I)/T,
     &      X(302,I)/T,X(303,I)/T,
     &      X(304,I)/T,X(305,I)/T,
     &      X(308,I)/T,X(309,I)/T,
     &      X(310,I)/T,TDAL(I)/T
          ELSE
            WRITE(7,6432) PHS(I),
     &      X(284,I)*F,X(285,I)*F,
     &      X(286,I)*F,X(287,I)*F,
     &      X(288,I)*F,X(289,I)*F*F2,
     &      X(290,I)*F*F3,X(291,I)*F,
     &      X(292,I)*F,X(293,I)*F,
     &      X(294,I)*F,X(295,I)*F,
     &      X(296,I)*F,X(297,I)*F,
     &      X(298,I)*F,X(299,I)*F,
     &      X(300,I)*F,X(301,I)*F,
     &      X(302,I)*F,X(303,I)*F,
     &      X(304,I)*F,X(305,I)*F,
     &      X(308,I)*F,X(309,I)*F,
     &      X(310,I)*F,TDAL(I)*F
          END IF
  101   CONTINUE
        CLOSE (UNIT=7)
      END IF
C
 252  FORMAT(   '       ALUMINUM SYSTEM IN ',A38)
 253  FORMAT(/T44,'    ALUMINUM SYSTEM IN ',A38//T50,'SOLID = ',A27)
 642  FORMAT(/,F5.2,1P2D12.4,3D12.4,D13.4,D14.4,2D12.4)
 6421 FORMAT(/,F5.2,  2F12.4,3F12.4,F13.4,F14.4,2F12.4)
 643  FORMAT(/,F5.2,1PD14.4,3D12.4,D14.4,3D12.4,D13.4)
```

```
Line# Source Line        Microsoft FORTRAN Optimizing Compiler Version 5.00

  848   6431 FORMAT(/,F5.2,  F14.4,3F12.4,F14.4,3F12.4,F13.4)
  849   6432 FORMAT(F5.2,26F10.4)
  850   6433 FORMAT(F5.2,26F6.1)
  851   644  FORMAT(/,F5.2,1PD13.4,2D15.4,5D12.4)
  852   6441 FORMAT(/,F5.2,  F13.4,2F15.4,5F12.4)
  853   171  FORMAT(/'  pH        Al(+3)     AlOH(+2)',
  854       & '  Al(OH)2(+)  Al(OH)3(0)  Al(OH)4(-)  Al2(OH)2(+4)',
  855       & '  Al3(OH)4(+5)  AlEDTA(-)  AlHEDTA(0)')
  856   172  FORMAT(/'  pH    AlOHEDTA(-2)   AlNTA(0)    AlHNTA(+)',
  857       & '  AlOHNTA(-)  Al(OH)2NTA(-2)  AlAC(+2)    AlCO3(+)',
  858       & '    AlSO4(+)  Al(SO4)2(-)')
  859   173  FORMAT(/'  pH    AlH2PO4(+2)  Al(H2PO4)2(+)  Al(H2PO4)3(0)'
  860       & '   AlA2(0)     AlF(+2)     AlF2(+)     AlF3(0)     T.D.Al')
  861  C
  862        RETURN
  863        END


WAL  Local Symbols

Name                   Class    Type              Size   Offset

MTLFILE . . . . . . . . . . . param                       0006
UNIT. . . . . . . . . . . . . param                       000a
T1. . . . . . . . . . . . . . param                       000e
TDAL. . . . . . . . . . . . . param                       0012
I . . . . . . . . . . . . . . local   INTEGER*4        4   0362
J . . . . . . . . . . . . . . local   INTEGER*4        4   0366
T . . . . . . . . . . . . . . local   REAL*8           8   036a
ISTART. . . . . . . . . . . . local   INTEGER*4        4   0372
Y . . . . . . . . . . . . . . local   REAL*8         208   0376
LABEL . . . . . . . . . . . . local   CHAR*38         38   0446
IEND. . . . . . . . . . . . . local   INTEGER*4        4   046c
SKIP. . . . . . . . . . . . . local   INTEGER*4        4   0470
PHS . . . . . . . . . . . . . OUT     REAL*4         164   98b8
X . . . . . . . . . . . . . . OUT     REAL*8      104632   0000
F . . . . . . . . . . . . . . W       REAL*8           8   0000
F2. . . . . . . . . . . . . . W       REAL*8           8   0008
F3. . . . . . . . . . . . . . W       REAL*8           8   0010
NUMPH . . . . . . . . . . . . W       INTEGER*4        4   0018
NPL . . . . . . . . . . . . . W       INTEGER*4        4   001c
PR. . . . . . . . . . . . . . W       INTEGER*4        4   003c
SLDID . . . . . . . . . . . . W       CHAR*27         27   0020

Global Symbols

Name                   Class    Type              Size   Offset

LOGFN . . . . . . . . . . . . extern  REAL*8         ***    ***
OUT . . . . . . . . . . . . . common  ***         104796   0000
```

```
Global Symbols

Name                      Class   Type              Size   Offset

W . . . . . . . . . . . . common  ***                 64     0000
WAL . . . . . . . . . . . FSUBRT  ***                ***     ab10
WCO . . . . . . . . . . . FSUBRT  ***                ***     7ffd
WFOOT . . . . . . . . . . extern  ***                ***      ***
WHEAD . . . . . . . . . . extern  ***                ***      ***
WK. . . . . . . . . . . . FSUBRT  ***                ***     6d01
WMNII . . . . . . . . . . FSUBRT  ***                ***     0000
WZN . . . . . . . . . . . FSUBRT  ***                ***     2926

Code size = e590 (58768)
Data size = 0a75 (2677)
Bss size  = 0474 (1140)

No errors detected
```

```
      SUBROUTINE WSO4(TDSO4,T1,UNIT,AFILE)
C
C     THIS BLOCK WRITES THE SULPHATE SUBSYSTEM AQUEOUS SPECIES REPORT.
      REAL*4 PHS(41),TDSO4(41)
      REAL*8 X(319,41),Y(24)
      COMMON /OUT/X,PHS
      INTEGER I,J,ISTART,IEND,SKIP,UNIT
      REAL*8 F,F2,F3,T,T1
      INTEGER NUMPH,NPL,PR
      CHARACTER LABEL*38,SLDID*27
      CHARACTER AFILE*15
      REAL*8 LOGFN
      COMMON /W/F,F2,F3,NUMPH,NPL,SLDID,PR
C
      ISTART=1
      IEND=0
      DO WHILE(IEND.NE.NUMPH)
        IEND=MIN0(IEND+NPL,NUMPH)
        CALL WHEAD
        IF (UNIT.EQ.1) LABEL='MOLES/L IN TERMS OF SPECIES           '
        IF (UNIT.EQ.2) LABEL='LOG(MOLES/L) IN TERMS OF SPECIES      '
        IF (UNIT.EQ.3) LABEL='%(MOLES) OF TOTAL IN TERMS OF SPECIES '
        IF (UNIT.EQ.4) LABEL='MG/L IN TERMS OF SULPHUR              '
        IF (UNIT.EQ.5) LABEL='LOG(MG/L) IN TERMS OF SULPHUR         '
        IF (UNIT.EQ.6) LABEL='%(MG) OF TOTAL IN TERMS OF SULPHUR    '
        WRITE(PR,201) LABEL,SLDID
        WRITE(PR,111)
        DO 11 I=ISTART,IEND
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            WRITE(PR,5991) PHS(I),
     &      LOGFN(X(  6,I)*F),LOGFN(X( 15,I)*F),
     &      LOGFN(X( 70,I)*F),LOGFN(X(143,I)*F),
     &      LOGFN(X( 10,I)*F),LOGFN(X(  8,I)*F),
     &      LOGFN(X(158,I)*F),LOGFN(X(159,I)*F*F2),
     &      LOGFN(X(150,I)*F),LOGFN(X(151,I)*F*F2)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            T=1
            IF (T1.NE.0D0) T=T1/100
            WRITE(PR,5991) PHS(I),
     &      X(  6,I)/T,X( 15,I)/T,
     &      X( 70,I)/T,X(143,I)/T,
     &      X( 10,I)/T,X(  8,I)/T,
     &      X(158,I)/T,X(159,I)*F2/T,
     &      X(150,I)/T,X(151,I)*F2/T
          ELSE
            WRITE(PR,599) PHS(I),
     &      X(  6,I)*F,X( 15,I)*F,
     &      X( 70,I)*F,X(143,I)*F,
     &      X( 10,I)*F,X(  8,I)*F,
     &      X(158,I)*F,X(159,I)*F*F2,
     &      X(150,I)*F,X(151,I)*F*F2
```

```
        END IF
   11 CONTINUE
      SKIP=2*(NPL+ISTART-IEND)
      CALL WFOOT(SKIP)
C
      CALL WHEAD
      WRITE(PR,201) LABEL,SLDID
      WRITE(PR,112)
      DO 12 I=ISTART,IEND
        IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
          WRITE(PR,5801) PHS(I),
     &    LOGFN(X(100,I)*F)   ,LOGFN(X(101,I)*F*F2),
     &    LOGFN(X( 46,I)*F)   ,LOGFN(X( 26,I)*F),
     &    LOGFN(X( 80,I)*F)   ,LOGFN(X(114,I)*F),
     &    LOGFN(X(128,I)*F)   ,LOGFN(X(129,I)*F*F2),
     &    LOGFN(X(130,I)*F*F3),LOGFN(X(147,I)*F)
        ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
          WRITE(PR,5801) PHS(I),
     &    X(100,I)/T   ,X(101,I)*F2/T,
     &    X( 46,I)/T   ,X( 26,I)/T,
     &    X( 80,I)/T   ,X(114,I)/T,
     &    X(128,I)/T   ,X(129,I)*F2/T,
     &    X(130,I)*F3/T,X(147,I)/T
        ELSE
          WRITE(PR,580) PHS(I),
     &    X(100,I)*F   ,X(101,I)*F*F2,
     &    X( 46,I)*F   ,X( 26,I)*F,
     &    X( 80,I)*F   ,X(114,I)*F,
     &    X(128,I)*F   ,X(129,I)*F*F2,
     &    X(130,I)*F*F3,X(147,I)*F
        END IF
   12 CONTINUE
      CALL WFOOT(SKIP)
C
      CALL WHEAD
      WRITE(PR,201) LABEL,SLDID
      WRITE(PR,113)
      DO 13 I=ISTART,IEND
        IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
          WRITE(PR,5991) PHS(I),
     &    LOGFN(X(230,I)*F),   LOGFN(X(300,I)*F),
     &    LOGFN(X(301,I)*F*F2),LOGFN(TDSO4(I)*F)
        ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
          WRITE(PR,5991) PHS(I),
     &    X(230,I)/T,X(300,I)/T,
     &    X(301,I)/T*F2,TDSO4(I)/T
        ELSE
          WRITE(PR,599) PHS(I),
     &    X(230,I)*F,X(300,I)*F,
     &    X(301,I)*F*F2,TDSO4(I)*F
        END IF
   13   CONTINUE
        CALL WFOOT(SKIP)
C
        ISTART=IEND+1
      END DO
C
      IF (AFILE(1:1).NE.' ') THEN
        OPEN (UNIT=7,FILE=AFILE,STATUS='UNKNOWN',
     &               RECL=250)
        WRITE(7,200) LABEL
        DO 71 I=1,NUMPH
          IF (UNIT.EQ.2.OR.UNIT.EQ.5) THEN
            Y(1 )=LOGFN(X(  6,I)*F)
            Y(2 )=LOGFN(X( 15,I)*F)
            Y(3 )=LOGFN(X( 70,I)*F)
            Y(4 )=LOGFN(X(143,I)*F)
            Y(5 )=LOGFN(X( 10,I)*F)
            Y(6 )=LOGFN(X(  8,I)*F)
            Y(7 )=LOGFN(X(158,I)*F)
            Y(8 )=LOGFN(X(159,I)*F*F2)
            Y(9 )=LOGFN(X(150,I)*F)
            Y(10)=LOGFN(X(151,I)*F*F2)
            Y(11)=LOGFN(X(100,I)*F)
            Y(12)=LOGFN(X(101,I)*F*F2)
            Y(13)=LOGFN(X( 46,I)*F)
            Y(14)=LOGFN(X( 26,I)*F)
            Y(15)=LOGFN(X( 80,I)*F)
            Y(16)=LOGFN(X(114,I)*F)
            Y(17)=LOGFN(X(128,I)*F)
            Y(18)=LOGFN(X(129,I)*F*F2)
            Y(19)=LOGFN(X(130,I)*F*F3)
            Y(20)=LOGFN(X(147,I)*F)
            Y(21)=LOGFN(X(230,I)*F)
            Y(22)=LOGFN(X(300,I)*F)
            Y(23)=LOGFN(X(301,I)*F*F2)
            Y(24)=LOGFN(TDSO4(I)*F)
            WRITE(7,5992) PHS(I),(Y(J),J=1,24)
          ELSE IF (UNIT.EQ.3.OR.UNIT.EQ.6) THEN
            WRITE(7,5993) PHS(I),
     &      X(  6,I)/T   ,X( 15,I)/T,
     &      X( 70,I)/T   ,X(143,I)/T,
     &      X( 10,I)/T   ,X(  8,I)/T,
     &      X(158,I)/T   ,X(159,I)*F2/T,
     &      X(150,I)/T   ,X(151,I)*F2/T,
     &      X(100,I)/T   ,X(101,I)*F2/T,
     &      X( 46,I)/T   ,X( 26,I)/T,
     &      X( 80,I)/T   ,X(114,I)/T,
     &      X(128,I)/T   ,X(129,I)*F2/T,
     &      X(130,I)*F3/T,X(147,I)/T,
     &      X(230,I)/T   ,X(300,I)/T,
     &      X(301,I)*F2/T,TDSO4(I)/T
```

```
Line#  Source Line          Microsoft FORTRAN Optimizing Compiler Version 5.00

  154              ELSE
  155                WRITE(7,5992) PHS(I),
  156         &      X(  6,I)*F   ,X( 15,I)*F,
  157         &      X( 70,I)*F   ,X(143,I)*F,
  158         &      X( 10,I)*F   ,X(  8,I)*F,
  159         &      X(158,I)*F   ,X(159,I)*F*F2,
  160         &      X(150,I)*F   ,X(151,I)*F*F2,
  161         &      X(100,I)*F   ,X(101,I)*F*F2,
  162         &      X( 46,I)*F   ,X( 26,I)*F,
  163         &      X( 80,I)*F   ,X(114,I)*F,
  164         &      X(128,I)*F   ,X(129,I)*F*F2,
  165         &      X(130,I)*F*F3,X(147,I)*F,
  166         &      X(230,I)*F   ,X(300,I)*F,
  167         &      X(301,I)*F*F2,TDS04(I)*F
  168              END IF
  169       71   CONTINUE
  170            CLOSE (UNIT=7)
  171          END IF
  172   C
  173    200   FORMAT(     '    SULPHATE SYSTEM IN ',A38)
  174    201   FORMAT(/T44,'    SULPHATE SYSTEM IN ',A38//T50,'SOLID = ',A27)
  175    599   FORMAT(/,F5.2,1P7D12.4,D13.4,2D14.4)
  176    5991  FORMAT(/,F5.2,  7F12.4,F13.4,2F14.4)
  177    580   FORMAT(/,F5.2,1PD12.4,2D13.4,4D12.4,D13.4,D14.4,D13.4)
  178    5801  FORMAT(/,F5.2,  F12.4,2F13.4,4F12.4,F13.4,F14.4,F13.4)
  179    5992  FORMAT(F5.2,24F10.4)
  180    5993  FORMAT(F5.2,24F6.1)
  181    111   FORMAT(/'  pH      S04(-2)     HS04(-)      ',
  182         & ' H2S04     NaS04(-)    MgS04(0)    CaS04(0)    ',
  183         & ' FeS04(+)    Fe(S04)2(-)    Hg2S04(0)   Hg2(S04)2(-2)')
  184    112   FORMAT(/'  pH    HgS04(0)   Hg(S04)2(-2)  ',
  185         & ' CdS04(0)    CuS04(0)    FeS04(0)    MnS04(0)    ',
  186         & ' ZnS04(0)   Zn(S04)2(-2)  Zn(S04)3(-4)    KS04(-)')
  187    113   FORMAT(/'  pH    CoS04(0)    AlS04(+)  Al(S04)2(-)     T.D.S04')
  188   C
  189          RETURN
  190          END


WS04  Local Symbols

Name                         Class   Type              Size   Offset

AFILE . . . . . . . . . . . . param                            0006
UNIT. . . . . . . . . . . . . param                            000a
T1. . . . . . . . . . . . . . param                            000e
TDS04 . . . . . . . . . . . . param                            0012
I . . . . . . . . . . . . . . local   INTEGER*4            4   0002
J . . . . . . . . . . . . . . local   INTEGER*4            4   0006
T . . . . . . . . . . . . . . local   REAL*8               8   000a
ISTART. . . . . . . . . . . . local   INTEGER*4            4   0012
```

```
WS04  Local Symbols

Name                          Class   Type              Size   Offset

Y . . . . . . . . . . . . . . local   REAL*8             192    0016
LABEL . . . . . . . . . . . . local   CHAR*38             38    00d6
IEND. . . . . . . . . . . . . local   INTEGER*4            4    00fc
SKIP. . . . . . . . . . . . . local   INTEGER*4            4    0100
PHS . . . . . . . . . . . . . OUT     REAL*4             164    98b8
X . . . . . . . . . . . . . . OUT     REAL*8          104632    0000
F . . . . . . . . . . . . . . W       REAL*8               8    0000
F2. . . . . . . . . . . . . . W       REAL*8               8    0008
F3. . . . . . . . . . . . . . W       REAL*8               8    0010
NUMPH . . . . . . . . . . . . W       INTEGER*4            4    0018
NPL . . . . . . . . . . . . . W       INTEGER*4            4    001c
PR. . . . . . . . . . . . . . W       INTEGER*4            4    003c
SLDID . . . . . . . . . . . . W       CHAR*27             27    0020


Global Symbols

Name                          Class   Type              Size   Offset

LOGFN . . . . . . . . . . . . extern  REAL*8             ***     ***
OUT . . . . . . . . . . . . . common  ***             104796    0000
W . . . . . . . . . . . . . . common  ***                 64    0000
WFOOT . . . . . . . . . . . . extern  ***                ***     ***
WHEAD . . . . . . . . . . . . extern  ***                ***     ***
WS04. . . . . . . . . . . . . FSUBRT  ***                ***    0000

Code size = 3750 (14160)
Data size = 025d (605)
Bss size  = 0104 (260)

No errors detected
```